Institute of Science and Technology
Tribhuvan University

Literature Review

On

Neural Machine Translation with Attention Mechanism

Submitted to

Central Department of Computer Science and Information Technology

Tribhuvan University

Submitted by

Joshana Shakya (15/075)

In partial fulfillment of the requirement for Master's Degree in Computer Science and Information Technology, 3rd Semester

November, 2021

Institute of Science and Technology
Tribhuvan University

Date: .....

Recommendation Letter of Supervisor

I hereby recommend that this literature review is prepared under my supervision by **Joshana Shakya** entitled **"Neural Machine Translation with Attention Mechanism"** be accepted as fulfillment in partial requirement for the degree of Master's of Science in Computer Science and Information Technology. In my best knowledge, this is an original work in computer science.

.....
Asst. Prof. Bal Krishna Subedi
Supervisor
CDCSIT, TU

CERTIFICATE OF APPROVAL

Date: .....

This is certify that the literature review prepared by Joshana Shakya entitled "Neural Machine Translation with Attention Mechanism"in partial fulfillment of the requirements for the degree of M.Sc. in Computer Science and Information technology has been well studied. In our opinion, it is satisfactory in the scope and quality as a project for the required degree.

.....

Asst. Prof. Bal Krishna Subedi

Supervisor

CDCSIT

TU

.....

Internal Examiner

.....

Asst. Prof. Nawaraj Paudel

Head of Department CDCSIT

TU

ACKNOWLEDGEMENT

It gives us immense pleasure to express my deepest sense of gratitude and sincere thanks to our highly respected and esteemed guide Asst. Prof. Bal Krishna Subedi for his valuable guidance, encouragement and help for completing this work. His useful suggestions for this whole work and co-operative behavior are sincerely acknowledged.

I would like to express my sincere thanks to our Department Head Asst. Prof. Nawaraj Paudel, Central Department of Computer Science and Information Technology, for whole hearted support.

I am also grateful to our teachers for their constant support and guidance. At the end I would like to express our sincere thanks to all my friends and others who helped me directly or indirectly during this seminar work.

Joshana Shakya (15/075)

ABSTRACT

Machine Translation is the computerized system for automatic translation from one language to another. Among different technologies for machine translation, the primary developments have been the rise of Neural Machine Translation. The neural machine translation system can be built using a sequence-to-sequence model with and without an attention mechanism. The vanilla sequence-to-sequence model stores sentence information of any length in a hidden vector of fixed size. On the other hand, the sequence-to-sequence model with an attention mechanism uses information about different parts of sentences. Both of these systems were designed using the TensorFlow platform to perform sentence-level translation from Nepali to English language; and BLEU score was used to evaluate the performance. The performance evaluation showed that the neural machine translation system translated better with sequence-to-sequence model using an attention mechanism.

Keywords: *Machine Translation, Neural Machine Translation, Sequence-to-Sequence Model, Attention Mechanism*

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

API	Application Programming Interface
ASCII	American Standard Code For Information Interchange
BLEU	Bilingual Evaluation Understudy
EBMT	Example-based Machine Translation
GPU	Graphics Processing Unit
LSTM	Long Short Term Memory
MT	Machine Translation
NMT	Neural Machine Translation
PBMT	Phrase-based Machine Translation
RNN	Recurrent Neural Network
SMT	Multi-label K-Nearest Neighbor
WMT	World Machine Translation

CHAPTER 1 INTRODUCTION

1.1 Introduction

The term "machine translation" (MT) refers to computerized systems responsible for an automatic translation from one language to another. The machine translation technology allows us to communicate effortlessly with those that speak a different language, or to understand the contents that are not in the users' native language. The several different types of MT approaches are rule-based, statistical, example-based, and neural. The progression in technologies has replaced the older systems with newer, more effective technologies. Up until late 2016 [9], the products using MT technology were hugely based on statistical methods known as Statistical Machine Translation (SMT). The SMT technology uses advanced statistical analysis to obtain the best possible translations for a word provided the context of a few surrounding words. Among different approaches to MT technology, the most significant developments have been the advent of Neural Machine Translation (NMT) [10]. Unlike the SMT technology, the NMT aims at building a single neural network that can be jointly tuned to obtain maximum translation performance [1].

Given a source sentence, $x_1, \ldots, x_n$, the neural machine translation system comprises of a neural network that directly models the conditional probability $p(y|x)$ of translating a source sentence to a target sentence, $y_1, \ldots, y_m$ [8]. In translation, the system maximizes the conditional probability, i.e., $\text{argmax}_y p(y|x)$ of the sentence pairs using a parallel training corpus. Once the distribution is learned, given a source sentence a corresponding translation can be generated by searching for the sentence that maximizes the conditional probability [1].

1.2 Sequence-to-Sequence Model

A sequence-to-sequence model is an end-to-end model consisting of two components:

i. Encoder

   An encoder reads the source sentence, a sequence of vectors $x = (x_1, \ldots, x_n)$ and

generates a fixed-dimensional representation s for the sequence. To accomplish this, the encoder reads the input tokens one at a time using a recurrent neural network cell such that

$$h_t = f(h_{t-1}, x_t) \tag{1.1}$$

and

$$s = q(\{h_1, \ldots, h_n\}) \tag{1.2}$$

where $h_t \in \mathbb{R}$ is a hidden state at time t, and s is a representation generated from the sequence of the hidden states. f and q are nonlinear functions. f is an LSTM and $q(\{h_1, \ldots, h_n\}) = h_n$ is the final hidden state of the cell.

ii. Decoder

The decoder is also an LSTM network whose hidden state of the first layer is initialized with the source representation s from the encoder. It is trained to to predict the next word y_j given the representation s and all the previously predicted words $\{y_1, \ldots, y_{j-1}\}$. That is, the decoder defines a probability over the translation y by decomposing the joint probability into the ordered conditionals:

$$\log p(y|x) = \sum_{j=1}^{m} \log p(y_j|y_{<j}, s) \tag{1.3}$$

The probability of decoding each word y_j is computed as:

$$p(y_j|y_{<j}, s) = \text{softmax}(g(h_j)) \tag{1.4}$$

where, g is the transformation function to produce a vocabulary-sized vector, h_j is the RNN hidden unit such that

$$h_j = f(h_{j-1}, s) \tag{1.5}$$

where f computes the current hidden state given the previous LSTM hidden state.

The token is appended to the end of the input to signify the decoder the start of the output generation. This is followed by a softmax on the final layer's output to generate the first output word. Then, that word is passed into the first layer to repeat the generation.

The graphical depiction of the model is shown in figure 1.1

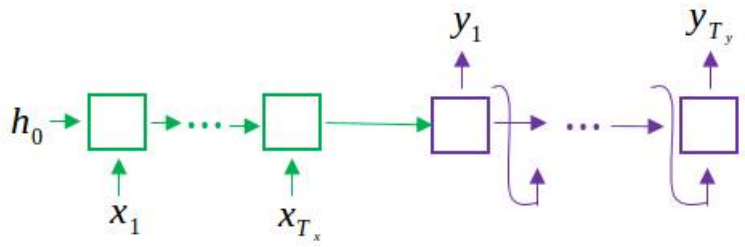

Figure 1.1: Sequence-to-Sequence Model [6].

The training objective of the sequence-to-sequence model is formulated as:

$$J_t = \sum_{(x,y)\in\mathbb{D}} -\log p(y|x) \tag{1.6}$$

with D being the parallel training corpus [8].

Once the output sequence is generated, the cross entropy loss of the model is minimized with a gradient descent algorithm and back-propagation. As both the encoder and decoder are trained at the same time, they both learn the same context vector representation [11] [1].

1.3 Attention Mechanism

Theoretically, a large enough and well-trained encoder-decoder model should be able to achieve perfect machine translation. However, in practice, the encoder-decoder attempts to store information about sentences of any arbitrary length in a hidden vector of fixed size. Thus, the network won't be able to encode all of the information in the longer sentences to translate [12]. The attention mechanism learns to assign significance to different parts of the input for each step of the output by providing the decoder network with a look at the entire input sequence at every decoding step; the decoder can then decide what input words are important at any point in time [11]. There are several types of attention mechanisms:

i. Bahdanau et al. NMT model

ii. Luong et al. NMT model

1.3.1 Luong et al. NMT model

In this model, during the decoding phase, at each time step t, the hidden state h_t is taken as input to derive the context vector c_t. Given the target hidden state h_t and the source-side context vector c_t, the model performs concatenation to combine the information from both vectors to produce attentional hidden state as follows:

$$\tilde{h_t} = \tanh(W_c[c_t; h_t]) \tag{1.7}$$

The attentional vector $\tilde{h_t}$is then fed through the softmax layer to produce the predictive distribution as:

$$p(y_t|y_{<t}, x) = \text{softmax}(W_s\tilde{h_t}) \tag{1.8}$$

The source-side context vector c_t can be computed in two ways:

i. Global Attention

ii. Local Attention

Global Attention

The global attention mechanism runs vanilla sequence-to-sequence model and considers all the hidden states of the encoder to derive the context vector c_t. A variable-length alignment vector of size equal to the number of time steps on the source side is derived using the current target hidden state h_t and each source hidden state $\bar{h_s}$ as follows [8]:

$$\begin{aligned} a_t(s) &= align(h_t, \bar{h}_s) \\ &= \frac{\exp(\text{score}(h_t, \bar{h}_s))}{\sum_{s'} \exp(\text{score}(h_t, \bar{h}_{s'})} \end{aligned} \tag{1.9}$$

Here, one of the following scoring functions is used:

$$\text{score}(h_t, \bar{h}_s) = \begin{cases} h_t^T \bar{h}_w & \text{dot} \\ h_t^T W_a \bar{h}_s & \text{general} \\ v_a^T \tanh(W_a[h_t; \bar{h}_s]) & \text{concat} \end{cases} \tag{1.10}$$

Given the alignment vectors as weights, the context vector c_t is computed as:

$$c_t = \sum_s a_t(s) \times h_s \tag{1.11}$$

This global attentional model is illustrated in figure 1.2.

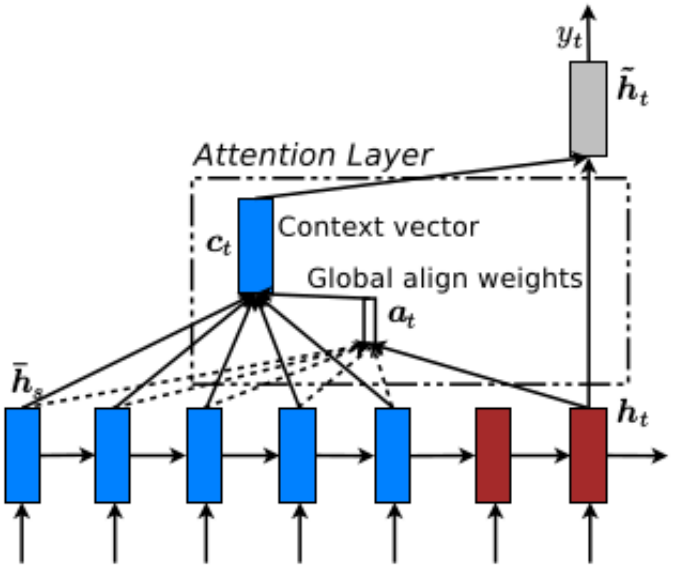

Figure 1.2: Global Attention Model [8].

1.4 Problem Statement

The neural machine translation system can be constructed with only vanilla sequence-to-sequence model. However, the encoder of such sequence-to-sequence model attempts to store sentences information of any arbitrary length in a hidden vector of fixed size. Thus, making the information incomplete for the decoder network. The attention mechanism provides the decoder the information about the different parts of the input for each step of the output.

1.5 Objective

The main objective of this literature review is to compare the performance of sequence-to-sequence model with and without attention mechanism in machine translation.

CHAPTER 2 LITERATURE REVIEW

2.1 Literature Review

The development of the commercial machine translation system started with rule-based machine translation in 70s. The rule-based machine translation system consists of bilingual dictionary and linguistic rules which parses text and creates a transitional representation to generate text in the target language [5] [17]. Later in 1984 [5], Makoto Nagao from Kyoto University came up with the notion of example-based machine translation (EBMT). The EBMT system is based on the existence of the large volume of the parallel bilingual texts that have been translated by professionals. In just five years after EBMT, the revolutionary invention of statistical translation was realized. The statistical machine translation (SMT) became the dominant framework of machine translation (MT) research as the system translated analyzing existing bilingual text corpora without using any rules and linguistics as a whole. During 2000, it became the dominant framework of MT research [5] [7]. In 2014 [5], the first scientific paper on using neural networks in machine translation was published.

The 2014 paper [1] of NMT introduced an extension to the encoder–decoder model to jointly learn to align and translate the sentence. The proposed model performs soft-search in a set of positions of a source sentence to obtain the most relevant information and generates a word in a translation. The model then predicts a target word utilizing the context vectors related with the source positions and the previously generated target words. The authors have shown that this model has improved translation performance compared to basic encoder-decoder model especially with longer sentences. They have performed the experiment with English-to-French translation using two types of models - RNN encoder-decoder and RNN search. They have trained each model twice: first with the sentences of length up to 30 words and then with the sentences of length up to 50 words. Their experiment revealed that the proposed model outperformed the conventional encoder-decoder model.

In [14], the authors have experimented on the powerful models of the deep neural networks to map sequences to sequences. They have used multi-layered LSTMs to map

the input sequence to the target output sequence. They have applied the model to the WMT'14 English to French MT task to directly translate the input sentence without using a reference SMT system. They have shown that a large deep LSTM with a limited number of vocabulary and no assumption about problem structure outperformed the standard SMT system with unlimited vocabulary; and have suggested that LSTM-based approach should perform well on other types of sequence learning problems.

In [13], the authors have explored different configurations to perform neural machine translation from English-to-Hindi language. They have used eight different architecture combinations of NMT and compared the corresponding results with conventional machine translation techniques. The experiment showed that the neural machine translation system provided better results for the larger dataset and when the number of layers in encoder and decoder are large. The authors have compared their attention based model with the SMT and PBMT systems; and concluded that the NMT system performed much better.

CHAPTER 3 METHODOLOGY

3.1 Data Cleaning and Preparation

The data is cleaned by first removing the punctuation characters. In the case of English text, the lowercase conversion of characters is performed; and the Unicode characters are normalized to ASCII characters. From the resulting data, non-alphabetic characters are removed. Finally, the vocabulary of each language is constructed with word index (mapping from word → id) and reverse word index (mapping from id → word). Each sentence is then padded to maximum length.

3.2 Neural Translation Model

An encoder and decoder model is used to construct a neural translation model. The LSTM variant of the recurrent neural network is used to construct each component of the model.

3.2.1 Recurrent Neural Networks

In recurrent neural networks, a model takes as input the current time step t and all of the previous time steps in order to compute the output $y^{<t>}$. At each time step t, the model is parameterized by shared weights W_{ax}, W_{aa} and W_{ya} , as shown in figure 3.1.

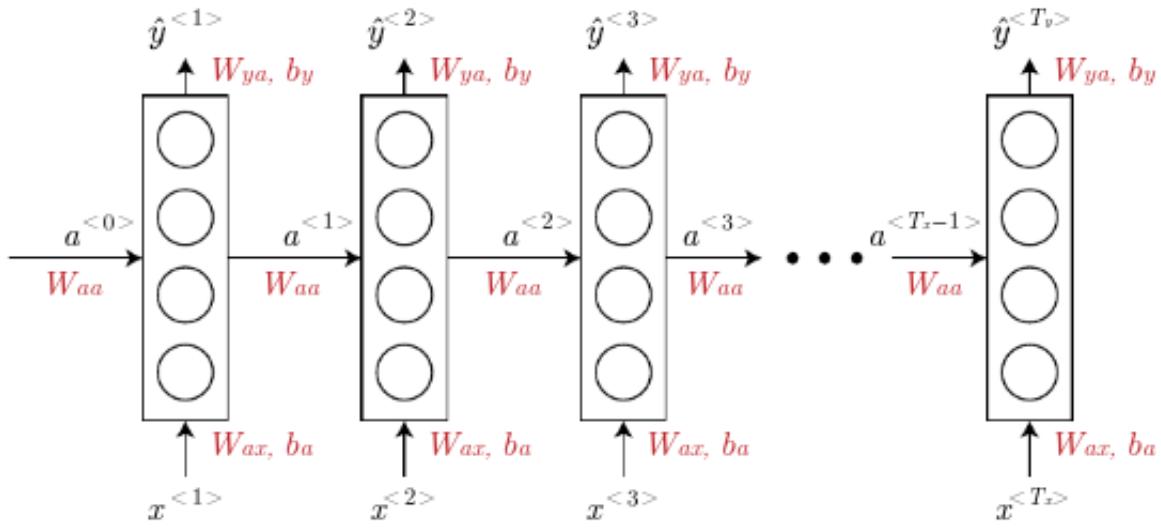

Figure 3.1: Structure of a basic recurrent neural network [3].

And the activation $a^{<t>}$ and the output $y^{<t>}$ are expressed as follows:

$$a^{<t>} = g(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a) \tag{3.1}$$

$$\hat{y}^t = g(W_{ya}^a t + b_y) \tag{3.2}$$

where g is any activation function.

Equation 3.1 is simplified as:

$$a^{<t>} = g(W_a[a^{<t-1>}, x^{<t>}] + b_a) \tag{3.3}$$

where W_a is formed by stacking

$$W_a = [W_{aa}|W_{ax}] \tag{3.4}$$

into a single matrix. The simplified version of equation 3.2 is written as:

$$\hat{y}^{<}t >= g(W_y a^{<t>} + b_y) \tag{3.5}$$

The loss at time step t is defined using cross-entropy as:

$$L^{<t>}(\hat{y}^{<t>}, y^{<t>}) = -y^{<t>} \log \hat{y}^{<t>} - (1 - y^{<t>}) \log(1 - \hat{y}^{<t>})) \tag{3.6}$$

For the entire sequence, the loss is computed as:

$$L(\hat{y}, y) = \sum_{t=1}^{T_y} L^{<t>}(\hat{y}^{<t>}, y^{<t>}) \tag{3.7}$$

The computation graph is pictured in figure 3.2.

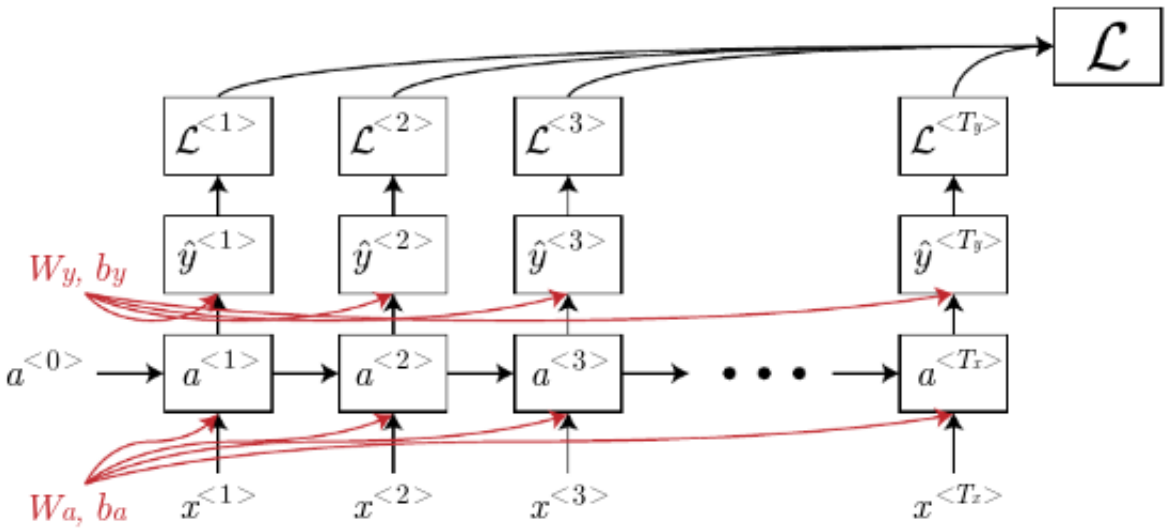

Figure 3.2: Loss computation graph [3].

3.2.2 Long Short Term Memory

Figure 3.3 shows an Long Short Term Memory (LSTM) unit.

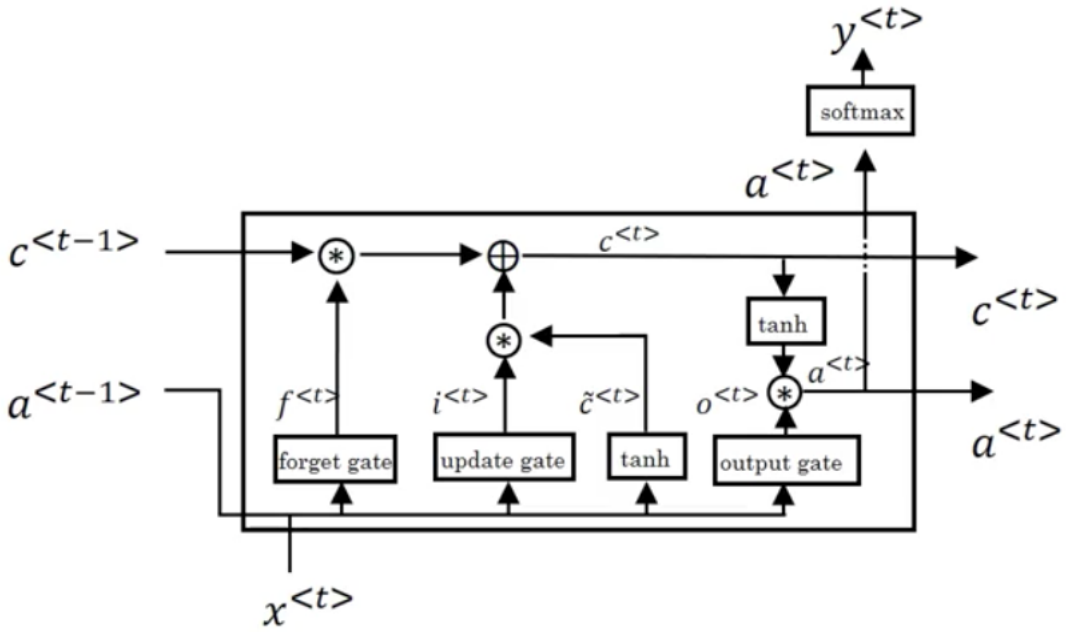

Figure 3.3: LSTM unit [3].

The 3 separate gates used in the model are forget gate Γ_f, update gate Γ_u, and output gate Γ_o, each of which are defined as:

$$\Gamma_u = \sigma(W_u \cdot [x_t, h_{t-1}]) \tag{3.8}$$

$$\Gamma_f = \sigma(W_f \cdot [x_t, h_{t-1}]) \tag{3.9}$$

$$\Gamma_o = \sigma(W_o \cdot [x_t, ht - 1]) \tag{3.10}$$

The other computations are:

$$\tilde{c}^t = \tanh(W_c[a^{<t-1>}, x^{<t>}] + b_c) \tag{3.11}$$

$$c^{<t>} = \Gamma_u \times \tilde{c}^{<t>} + \Gamma_f \times c^{<t-1>} \tag{3.12}$$

$$a^{<t>} = \Gamma_o \times \tanh(c^{<t>}) \tag{3.13}$$

Here, $\tilde{c}^{<t>}$ is new information and $c^{<t-1>}$ is the old memory cell information [3].

3.3 Translation

To generate translations from a probability model, the Greedy 1-best search criterion is used. In greedy search, pt at every time step is calculated and the word that gives the highest probability is selected to use as the next word in our sequence. In other words,

$$x_t = \text{argmax}_{\tilde{x}_t} \mathbb{P}(\tilde{x}_t | x_1, \ldots, x_n) \tag{3.14}$$

This technique is efficient and natural, however, it explores a small part of the search space and if there is a mistake at one-time step, the rest of the sentence could be heavily impacted [11].

CHAPTER 4 EXPERIMENTATION

4.1 Dataset

The initial requirement for carrying out the experiment is the availability of parallel corpus for source and target languages. Nepali is not a resourceful language as there is no huge availability of large datasets. The following two different datasets are considered for the experiments.

i. English-Nepali Parallel Corpus from the Nepali National Corpus [4]

ii. tico-19 v2020-10-28 from the Translation Initiative for COVID-19 [16]

The following table gives the details of the datasets used.

Table 4.1: Details of datasets

Corpus	English-Nepali Parallel Corpus			tico-19 v2020-10-28
	project_save	NP2	NP8	
No. of sentences	1060	50	1054	3070

The sentences with length greater than 30 are discarded for the experiment. The final collection consists of 3981 sentence pairs. 3000 sentence pairs are used as training samples and the remaining 981 sentence pairs are used as testing samples.

4.2 Implementation Environment

The simple NMT system is implemented by taking reference from the tutorials by Jason Brownlee [2] and the NMT with attention is developed using TensorFlow Addons Networks : Sequence-to-Sequence NMT with Attention Mechanism as a guide [15].

Anaconda distribution of Python 3.7.12 is used as programming language. TensorFlow 2.6.0 as platform and TensorFlow Addons 0.11.2 as API are used in the system design. The final experiment is carried out on Google Colab with GPU as runtime environment.

4.3 Parameters

The following parameters are used to carry out the experiment.

Table 4.2: Parameters

No. of epochs	500
Batch size	64
No. of units	256
Activation	softmax
Optimizer	Adam
Default learning rate	0.001
Loss	categorical cross entropy
embedding_dim	128 and 256
No. of layers	unknown

4.4 Sample Translations

The sample translations in Nepali to English direction using simple NMT and NMT with attention are respectively shown in Table 4.3 and Table 6.

Table 4.3: Simple NMT Nepali-English Translation

Training	
src	हामीलाई परोपकारी अर्थशास्त्रको जरुरत छ
ref	we need caring economics
pred	the the
Testing	
src	सरकारी क्षेत्रको ऋणात्मक बचत र लगानीको अन्तरलाई विदेशी सहायता र आन्तरिक ऋणबाट बेहोरिनेछ।
ref	the deficit between the public sector negative savings and investment will be borne thorough foreign assistance and internal borrowing
pred	the the the the the the the the the

Table 4.4: NMT with Attention Nepali-English Translation

Training	
src	सहरमा निजी वाहन प्रयोग गर्न प्रतिबन्ध लगाइएको थियो।
ref	private vehicle use was banned in the city
pred	private vehicle use was banned in the export
Testing	
src	चरण ii परीक्षणहरू प्रभावकारिताको प्रारम्भिक पठन स्थापना गर्न प्रयोग गरिन्छन् र nce द्वारा लक्षित रोग भएका मानिसहरूको थोरै सङ्ख्यामा थप सुरक्षा अन्वेषण गरिन्छ।
ref	phase ii trials are used to establish an initial reading of efficacy and further explore safety in small numbers of people having the disease targeted by the nce
pred	testing of example respiratory providers is after after after the spread of a small virus that which is available after a significant risk of infection

CHAPTER 5 RESULT AND ANALYSIS

5.1 Performance Evaluation

5.1.1 Bilingual Evaluation Understudy

The Bilingual Evaluation Understudy (BLEU) algorithm evaluates the precision score of a candidate machine translation against a reference human translation. The reference translation is a assumed to be a model example of a translation. The algorithm identifies all of n-gram matches and evaluates the strength of the match with the precision score. The precision score is the fraction of n-grams in the translation that also appear in the reference.

Let k be the maximum n-gram to evaluate the score of translation. Let

$$p_n = \frac{\#\text{ matched } n\text{-grams}}{\#\ n\text{-grams in candidate translation}} \tag{5.1}$$

the precision score for the grams of length n.

Finally, let $w_n = \frac{1}{2n}$ be a geometric weighting for the precision of the n'th gram. The brevity penalty is defined as:

$$\beta = \exp^{min\left(0,1-\frac{len_{ref}}{len_{MT}}\right)} \tag{5.2}$$

where len_{ref} is the length of the reference translation and len_{MT} is the length of the machine translation.

The BLEU score is then defined as [11]:

$$\text{BLEU} = \beta \prod_{i=1}^{k} p_n^{w_n} \tag{5.3}$$

5.2 Result and Analysis

The BLEU score on training data is shown in Table 5.1.

Table 5.1: BLEU score on training data

	Simple NMT	**NMT with Attention**
BLEU-1	0.031293	0.825670
BLEU-2	0.089161	0.801728
BLEU-3	0.135540	0.805319
BLEU-4	0.150501	0.774398

The BLEU score on test data is shown in Table 5.2.

Table 5.2: BLEU score on test data

	Simple NMT	**NMT with Attention**
BLEU-1	0.030298	0.378832
BLEU-2	0.086331	0.306991
BLEU-3	0.131241	0.315862
BLEU-4	0.145728	0.262754

The BLEU scores show that NMT with Attention outperformed the simple NMT system in Nepali to English sentence level translation.

CHAPTER 6 CONCLUSION

6.1 Conclusion

In this literature review, sentence-level Nepali to English neural machine translation with and without attention mechanism was studied. The experiments were carried out on the collection of two datasets - English-Nepali Parallel Corpus from the Nepali National Corpus and tico-19 v2020-10-28 from the Translation Initiative for COVID-19. Out of 3981 pairs of sentences, 3000 pairs were used for training, and the remaining 981 pairs for testing. The performance of the systems was evaluated using the BLEU score.

From the observations, it can be concluded that neural machine translation system works much better when the attention mechanism was included.

6.2 Future Recommendation

As of now, this neural machine translation system has become obsolete. So, this study can be carried out using the Transformer mechanism.

REFERENCES

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

[2] Brownlee, J., "How to develop a neural machine translation system from scratch," 2020. [Online]. Available: https://machinelearningmastery.com/develop-neural-machine-translation-system-keras/.

[3] Deitke, M., "Sequence models," 2020. [Online]. Available: https://mattdeitke.com/notes/cs230#pf5b.

[4] Duwal, S., Manandhar, A., Maskey, S., and Hada, S., "Nepali translator," 2019.

[5] freeCodeCamp.org, "A history of machine translation from the cold war to deep learning," 2018. [Online]. Available: https://www.freecodecamp.org/news/a-history-of-machine-translation-from-the-cold-war-to-deep-learning-f1d335ce8b5/.

[6] Giorgi, J., "Week 3: Sequence models & attention mechanism." [Online]. Available: https://johngiorgi.github.io/deeplearning.ai-coursera-notes/sequence_models/week_3/.

[7] Hutchins, J., "The history of machine translation in a nutshell," *Retrieved December*, vol. 20, no. 2009, pp. 1–1, 2005.

[8] Luong, M.-T., Pham, H., and Manning, C. D., "Effective approaches to attention-based neural machine translation," *ArXiv preprint arXiv:1508.04025*, 2015.

[9] "Machine translation." [Online]. Available: https://www.semantix.com/machine-translation [Accessed: 11/11/2021].

[10] "Machine translation: A comprehensive guide: Memsource," 2021. [Online]. Available: https://www.memsource.com/blog/a-comprehensive-guide-to-machine-translation/.

[11] Manning, C., Socher, R., Fang, G. G., and Mundra, R., "Cs224n: Natural language processing with deep learning," 2019.

[12] Neubig, G., "Neural machine translation and sequence-to-sequence models: A tutorial," *ArXiv preprint arXiv:1703.01619*, 2017.

[13] Saini, S. and Sahula, V., "Neural machine translation for english to hindi," in *2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP)*, IEEE, 2018, pp. 1–6.

[14] Sutskever, I., Vinyals, O., and Le, Q. V., "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[15] "Tensorflow addons networks : Sequence-to-sequence nmt with attention mechanism." [Online]. Available: https://www.tensorflow.org/addons/tutorials/networks_seq2seq_nmt.

[16] Tiedemann, J., "Parallel data, tools and interfaces in opus.," in *Lrec*, Citeseer, vol. 2012, 2012, pp. 2214–2218.

[17] "What is machine translation? rule based vs. statistical." [Online]. Available: https://www.systransoft.com/systran/translation-technology/what-is-machine-translation/.