English to Nepali Translation

Wang Dorjee Sherpa [1] Prof. Chinmoy Ghosh[2]

***Abstract*— Neural Machine Translation , or NMT for short, is the use of neural network models to learn statistical model for machine translation. The key benefit to the approach is that a single system can be trained directly on source and target text, no longer requiring the pipeline of specialised systems used in statistical machine learning. "Unlike the traditional phrase-based translation system which consists of many small sub-components that are tuned separately, neural machine translation attempts to build and train a single, large neural network that reads a sentence and outputs a correct translation." As such, neural machine translation systems are said to be end-to-end systems as only one model is required for the translation. In this project, I have used Sequence-to-Sequence models with different architectures and training method.**

1) **First version: Simple sequence-to-sequence model**
2) **Second version: Sequence-to-Sequence (teacher forcing)**
3) **Third version: Same as second version but with pre-trained glove embeddings**
4) **Fourth version: Sequence-to-sequence with Attention Mechanism**

I. INTRODUCTION

One of the earliest goals for computers was the automatic translation of text from one language to another. Automatic or machine translation is perhaps one of the most challenging artificial intelligence tasks given the fluidity of human language. Classically, rule-based systems were used for this task, which were replaced in the 1990s with statistical methods. More recently, deep neural network models achieve state-of-the-art results in a field that is aptly named neural machine translation.

A. What is Machine Translation?

Machine translation is the task of automatically converting source text in one language to text in another language. Given a sequence of text in a source language, there is no one single best translation of that text to another language. This is because of the natural ambiguity and flexibility of human language. This makes the challenge of automatic machine translation difficult, perhaps one of the most difficult in artificial intelligence.

B. What is Statistical Machine Translation?

Statistical machine translation, or SMT for short, is the use of statistical models that learn to translate text from a source language to a target language given a large corpus of examples. The approach is data-driven, requiring only a corpus of examples with both source and target language text. This means linguists are not longer required to specify the rules of translation.

Although effective, statistical machine translation methods suffered from a narrow focus on the phrases being translated, losing the broader nature of the target text. The hard focus on data-driven approaches also meant that methods may have ignored important syntax distinctions known by linguists. Finally, the statistical approaches required careful tuning of each module in the translation pipeline.

C. What is Neural Machine Translation?

Neural machine translation, or NMT for short, is the use of neural network models to learn a statistical model for machine translation. The key benefit to the approach is that a single system can be trained directly on source and target text, no longer requiring the pipeline of specialized systems used in statistical machine learning. As such, neural machine translation systems are said to be end-to-end systems as only one model is required for the translation.

D. Encoder-Decoder Model (sequence-to-sequence)

Multilayer Perceptron neural network models can be used for machine translation, although the models are limited by a fixed-length input sequence where the output must be the same length. These early models have been greatly improved upon recently through the use of recurrent neural networks organized into an encoder-decoder architecture that allow for variable length input and output sequences. Key to the encoder-decoder architecture is the ability of the model to encode the source text into an internal fixed-length representation called the context vector. Interestingly, once encoded, different decoding systems could be used, in principle, to translate the context into different languages.

E. Encoder-Decoders with Attention

Although effective, the Encoder-Decoder architecture has problems with long sequences of text to be translated. The problem stems from the fixed-length internal representation that must be used to decode each word in the output sequence. The solution is the use of an attention mechanism that allows the model to learn where to place attention on the input sequence as each word of the output sequence is decoded.

The encoder-decoder recurrent neural network architecture with attention is currently the state-of-the-art on some benchmark problems for machine translation. And this architecture is used in the heart of the Google Neural Machine Translation system, or GNMT, used in their Google Translate service.

*This work was not supported by any organization

[1]Bachelor of Technology in Computer Science and Engineering, JALPAIGURI GOVERNMENT ENGINEERING COLLEGE
Wang Dorjee Sherp
[2] Assistant Professor, Department of Computer Science and Engineering
ghosh at jalpaiguri

Although effective, the neural machine translation systems still suffer some issues, such as scaling to larger vocabularies of words and the slow speed of training the models. There are the current areas of focus for large production neural translation systems, such as the Google system.

F. Project Goal

In this project, we will develop a neural machine translation system for translating English phrases to Nepali. The idea is to feed an English phrase to a model and get its equivalent translation in Nepali language. We will use four different models for this task from a very basic Seq2Seq model to Seq2Seq model with attention.

G. Model Details

1) First Version: Simple Seq2Seq Model: We feed in the input sequence, which first goes through the encoder (an embedding layer followed by a single LSTM layer), which outputs a vector, then it goes through a decoder (a single LSTM layer, followed by a dense output layer), which outputs a sequence of vectors, each representing the estimated probabilities for all possible output character. Since the decoder expects a sequence as input, we repeat the vector (which is output by the decoder) as many times as the longest possible output sequence.

2) Second Version: Seq2Seq with Teacher Forcing Method: Feeding the shifted targets to the decoder (teacher forcing). Instead of feeding the decoder a simple repetition of the encoder's output vector, we can feed it the target sequence, shifted by one time step to the right. This way, at each time step the decoder will know what the previous target character was. This should help to tackle more complex sequence-to-sequence problems. Since the first output character of each target sequence has no previous character, we will need a new token to represent the start-of-sequence ($start\_$). During inference, we won't know the target, so what will we feed the decoder? We can just predict one character at a time, starting with an $start\_$ token, then feeding the decoder all the characters that were predicted so far. But if the decoder's LSTM expects to get the previous target as input at each step, how shall we pass it the vector output by the encoder? Well, one option is to ignore the output vector, and instead use the encoder's LSTM state as the initial state of the decoder's LSTM (which requires that encoder's LSTM must have the same number of units as the decoder's LSTM).

3) Third Version: Seq2Seq with Pre-trained Glove Embedding: This model is similar to the previous model. Only the difference is that we will use pre-trained glove embeddings for encoder's embedding layer. This way the model will no longer have to learn embeddings for English words.

4) Fourth Version: Seq2Seq Model with Attention: Here we will use Bahdanau Attention mechanism also known as Additive attention as it performs a linear combination of encoder states and the decoder states. Attention mechanism suggested by Bahdanau

1) All hidden states of the encoder(forward and backward) and the decoder are used to generate the context vector, unlike how just the last encoder hidden state is used in seq2seq without attention.
2) The attention mechanism aligns the input and output sequences, with an alignment score parameterized by a feed-forward network. It helps to pay attention to the most relevant information in the source sequence.
3) The model predicts a target word based on the context vectors associated with the source position and the previously generated target words.

In this model, the decoder decides which part of the source sentence it needs to pay attention to, instead of having encoder encode all the information of the source sentence into a fixed-length vector

II. Related Works

NLP has many applications but machine translation is considered as on its earliest application. Globally it started around the year 1959, but in India it reached in the year 1980. Since then, many institutions and organizations are working on MT. Among them, the eminent institutes are IIT Kanpur, C-DAC Mumbai, University of Hyderabad, C-DAC Pune, TDIL, etc. In 2003, bharati and his team demonstrated a domain free MT system, named "Anusaaraka" that is designed in the year 1995 at IIT Kanpur [5]. Though it is a domain free system but the system mainly used for translating children's stories. In the year 1999 at C-DAC, Bangalore, "Mantra"—a machine translation system was developed mainly for the Rajya Sabha Secretariat. Nowadays, it also works for any Indian language pairs [6]."Matra", another MT system has been developed in the year 2004 at C-DAC. It translates an English sentence to its equivalent Hindi sentence. It is a domain free system [7].Using the concept of rule-based approach and the generalized form of the lexicon, a MT system named "AnglaBharti" was designed. It has been developed in the year 1991 at IIT Kanpur. After "AnglaBharti", another MT system named 'AnuBharti' was developed in the 2004 at same institution [8]. In the year 2004 at Jadavpur University Kolkata, a machine translation system "Anubaad" was developed for common Bengali people those who don't know English. The system mainly translates English news to its equivalent Bengali news [9].Sampark machine translation team of Consortium of Institutions has been designed a machine translation system named "Sampark" in the year 2009. It works for almost all language pairs [10].

III. Proposed Methodology / Implementation

A. Plan of Work

1) Data Collection:: The dataset was created using the English to Spanish dataset. The English phrases were extracted from English-to-Spanish dataset and translated with the help of Google Translate to Nepali. This process was automated using the selenium library and translated almost 10,000 phrases.

2) Data Preprocessing: The dataset was preprocessed in the following way

For English texts

- Remove punctuations
- Convert all words to lower case
- Remove digits
- Subword tokenisation (don't -> do not)

For Nepali texts

- Remove digits
- Remove sentences containing english words if exists

Data Splitting

- Total: 9886
- Training: 7908
- Validation: 989
- Test: 989

Tools & Technology Used

- Python
- TensorFlow
- Keras
- Pandas
- NumPy
- Google Colab

The objective is to feed the English sentences to the model and get its equivalent translation in Nepali. Only sequences with maximum 11 words is used in this project. In short, English sentences are fed to the encoder, and the decoder outputs the Nepali translation. Note that the Nepali sentences are also used as inputs to the decoder, but shifted back by one step. In other words, the decoder is given the word that it should have output at the previous step (regardless of what it actually output). For the very first word, it is given the start-of-sequence ($start\_$) token. The decoder is expected to end the sentence with and end-of-sentence ($end\_$) token.

B. Current status of the project development

So far all models have been trained and evaluated using BLEU (Bilingual Evaluation Understudy) score.

IV. CONCLUSION

We developed a neural machine translation system for translating English phrases to Nepali. We collected the data, preprocessed it and successfully trained different models out of which attention based model performed the best as expected. The model we trained is still not sufficient to translate all the phrases we want. But it can be further improved by exploring other techniques or using transformer based models.

V. FUTURE WORK

1) Use Beam Search for inference.
2) The dataset used to fit the model could be expanded to 50,000, 100,000 phrases, or more.
3) The model could use regularization, such as weight or activation regularization.
4) The encoder and/or the decoder models could be expanded with additional layers and trained for more epochs, providing more representational capacity for the model.
5) The order of input phrases could be reversed, which has been reported to lift skill,or a Bidirectional input layer could be used.

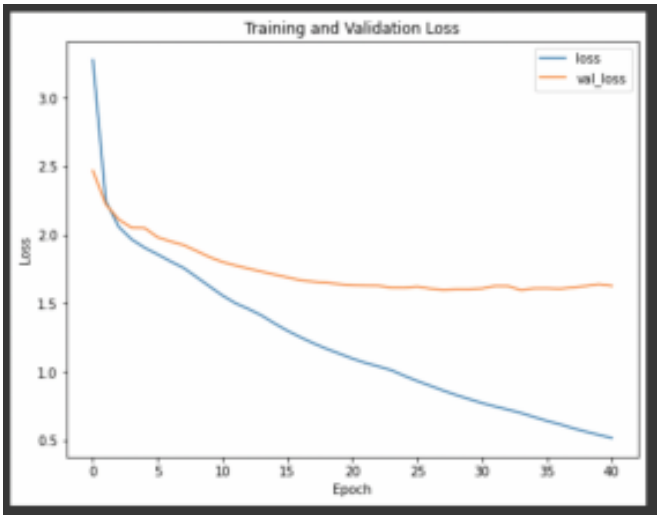

Fig. 1. Model 1: Loss plot

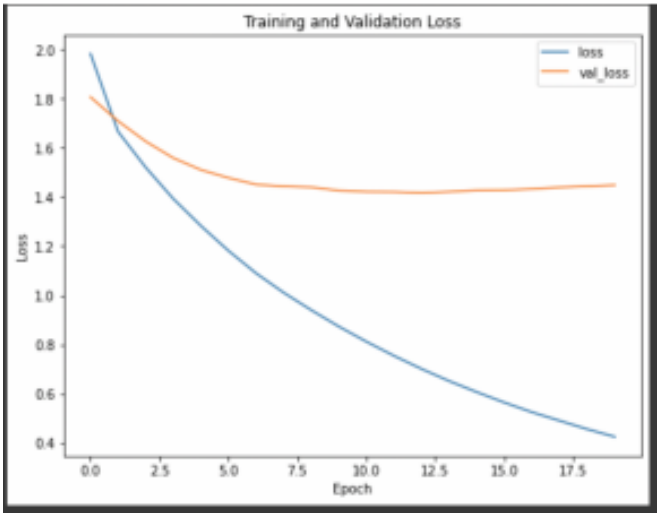

Fig. 2. Model 2: Loss plot

VI. LIST OF FIGURES

- Figure 1 shows a Model 1 of Loss plot.
- Figure 2 shows a Model 2 of Loss plot.
- Figure 3 shows a Model 3 of Loss plot.
- Figure 4 shows a Model 4 of Loss plot.

VII. LIST OF TABLES

ACKNOWLEDGMENT

The project work summarized in this report, explores the topic named: "English to Nepali Translation" which has been trained using Sequence-to-Sequence models.

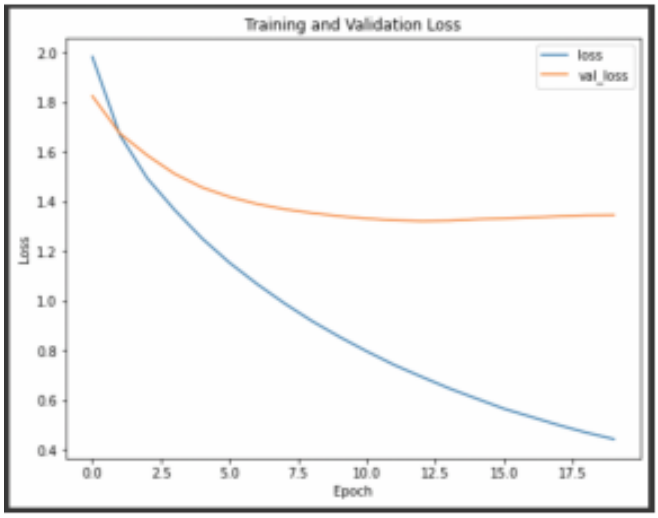

Fig. 3. Model 3: Loss plot

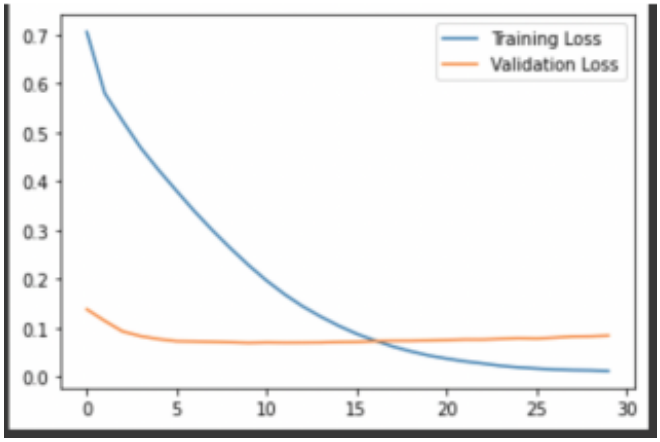

Fig. 4. Model 4: Loss plot

TABLE I

LOSS TABLE

Model Version	Training Loss	Validation Loss	Test Loss
FIRST	0.5178	1.6269	1.5966
SECOND	0.4251	1.4486	1.4134
THIRD	0.4417	1.3442	1.3144
FOURTH	0.0119	0.0843	0.0862

TABLE II

BLEU SCORE TABLE

Model Version	BLEU-1	BLEU-2	BLEU-3	BLEU-4
FIRST	0.2081	0.0742	0.0344	0.0099
SECOND	0.2401	0.0993	0.0534	0.0152
THIRD	0.2790	0.1280	0.0777	0.0334
FOURTH	0.4772	0.3363	0.2887	0.1783

We hereby thank our project supervisor Prof. Chinmoy Ghosh. This is a combined endeavour of a number of people who directly or indirectly helped us in completing our project synopsis. A word of thanks also goes to all our friends for being our best critics. And finally, this documentation would never have been more educative and efficient without the constant help and guidance of our project guide Prof. Chinmoy Ghosh. We would like to thank him for giving us the right guidance and encouraging us to complete the project within time. We also express our deepest and sincere gratitude to all our teachers for their kind comments and advice for our project. Last but not the least, we would also like to express our heartiest gratitude to our HoD, Prof. Subhas Barman, Prof. Sambhu Nath Pattanaik and other faculties for their encouragement and kind suggestion.

REFERENCES

[1] Aurelien Geron, “Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow” (Book)
[2] Francois Chollet, The Keras Blog
[3] TensorFlow Documentation
[4] Bharati, A., Chaitanya, V., Kulkarni, A. P., Sangal, R. (2003). Anusaaraka: machine translation in Stages. arXiv preprint cs/0306130.
[5] Dwivedi, S. K., Sukhadeve, P. P. (2010). Machine translation system in indian perspectives. Journal of computer science, 6(10), 1111.
[6] Rao, D. (2001). Machine translation in India: A brief survey. In Proceedings of SCALLA 2001 Conference,(SCALLA’01), National Centre for Software Technology. Banglaore, India (pp. 1-6).
[7] Sinha, R. M. K., Jain, R., Jain, A. (2001, February). Translation from English to Indian languages: Anglabharti approach. In Proceeding of the Symposium on Translation Support System, Feb (pp. 15-17).
[8] Bandyopadhyay, S. (2000). ANUBAAD-the translator from English to Indian languages. In Proceedings of the 7th State Science and Technology Congress,(SSTC’00), Calcutta, India (pp. 1-9).
[9] Dwivedi, S. K., Sukhadeve, P. P. (2010). Machine translation system in indian perspectives. Journal of computer science, 6(10), 1111.