

Attention Based Neural Machine Translator for Hindi to English Language

Abhishek Chaudhary¹, Akhil kumar verma², Dr. Shubham Shukla³

¹²³Electronics and Communication Engineering, KIET Group of Institution, Ghaziabad, India

Abstract: -Neural machine translation, or NMT uses the models based on neural network to gain an understanding of the statistical model so that it can be used in machine translation. Attention based mechanism used in neural networks make the translation process more reliable by using the bidirectional approach of encoder and decoder and is used to compare BLEU from different sources

Keywords— NLP, Attentional Mechanism, Encoder, Decoder, Machine Translations (MT), Recurrent Neural Network

I. INTRODUCTION

One of the earliest goals of the computer was the automatic translation of text from one language to another. Machine translation is conceivably one of the most demanding artificial intelligence tasks given the fluidity of human being language. Machine translation is the task of spontaneously converting source text in one language to text into a different language. MT performs as a channel for cross-language interaction in Natural Language Processing (NLP). MT manages language confusion problems by using computerized translation in between dual languages while keeping its meaning intact. The attentional mechanism has recently occurred to prove to be useful in neural machine translation by selectively concentrating on elements of the trace sentence during translation. NMT is fascinating since it needs marginal domain knowledge to get hands on and has the power to simplify well to very extended word sequences. For doing this we have used encoder and decoder which uses Recurrent Neural network Architecture.

In this paper, we create, with minimalism and efficiency in attention, one type of attention-based model that is the approach in which all provided words given to be attended specifically using the bhadanau model.

II. NEURAL MACHINE TRANSLATOR

The neural machine translator is a neural based network that promptly models the conditional probability $p(b|a)$ of converting a trace sentence, a_1, \dots, a_n , to a end sentence, b_1, \dots, b_m . It comprised of two elements i.e encoder and decoder. an encoder computes a description k for each trace sentence and the decoder which produces one end word at a time and hence crumbles the conditional probability as: $\log p(b|a) = \sum_{j=1}^m \log p(b_j | b_{<j}, k)$. In this paper, following (Sutskever et al., 2014; Luong et al., 2015), the use of assembling LSTM architecture has been taken for building the NMT systems, as illustrated in figure

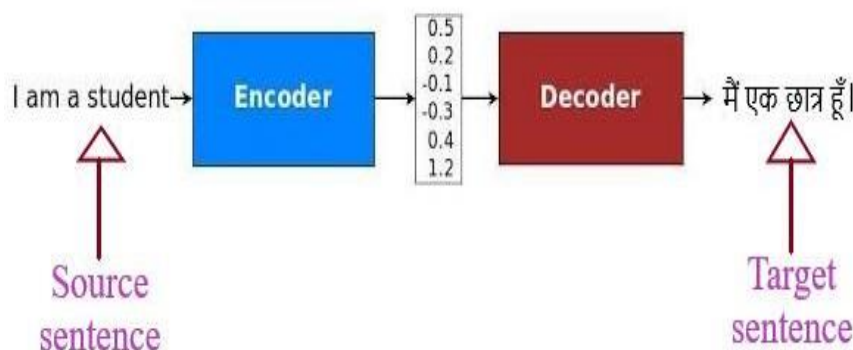


Fig. 1 Encoder and decoder in NMT

III. ATTENTION BASED MODEL

The attention model to be used is type of global that differs from the local in the format whether the “attention” is positioned on all trace positions or on only a limited trace position. It is to be noticed that at each time step t in the decryption phase the approach first take input as the hidden state h_t at the upper layer of a assembling LSTM so that to obtain a context vector c_t that gets significant trace-side data to help calculate the current end word b_t or to be said in simpler terms is to contemplate all the hidden states of the encoder when stemming the context vector c_t . Also, in addition to what we did differently than already propose bhadanau model is that we use the chain of the forward and backward trace hidden states in the bi-directional encoder.

IV. METHODOLOGY

Step 1: Import all the necessary python libraries-TensorFlow 2.8, Sklearn, Matplotlib, Numpy and Pandas.

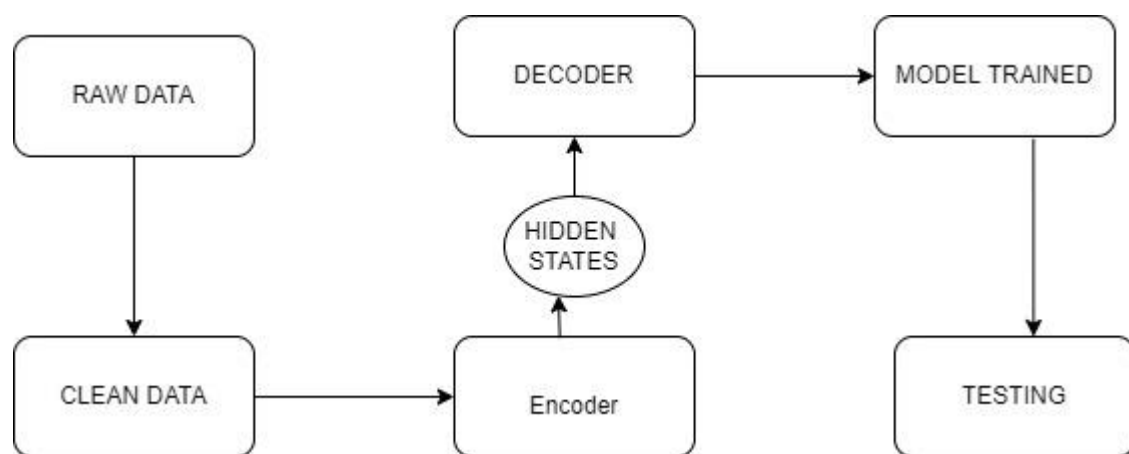
Step 2: Data Collection

Step 3: Pre-processing (Cleaning) and standardization

Step 4: Writing the scripts for decoder and encoder

Step 5: Train the model with pre-processed dataset.

Step 6: Testing of the model with sample inputs.



V. DATA COLLECTION

Although there was not much dataset for Hindi to English on the internet, but we found out a dataset posted on Kaggle which consisted of more than 3000 sentences, but it consisted of various adds on with respect to the Hindi and English translation which is needed to be removed in order to process data. We also looked for the dataset provided on TensorFlow, but it too has the same ambiguity.

VI. PRE-PROCESSING OF DATA

The main problem is that model is dealing with bilingual text with a restricted vocabulary. So, it was essential to standardize the stored text that is to be trained. Firstly, we Unicode normalization to split stressed characters and substitute compatibility characters with their ASCII counterparts which is done by tensor flow-text package. Then this standardization function will be enfolded up in a text vectorization layer which will grip the vocabulary mining and translation of input text to sequences of tokens. Remove multiple space that was generated while data collection. This can cause Vocabulary wrong interpretation, so the next target was to remove these extra spaces. Remove stop-words as Stop-words are those words that generally do not contain important information. Splitting of the accented characters, adding spaces around punctuation and strip whitespace.

The Text Vectorization layer and several other preprocessing layers have adapt method. This technique read out one epoch of the training data and works a lot like Model Fix Method. This adapt method sets the layer centred on the data. Here it

determines the vocabulary. The refunded token IDs are null padded. This can simply be rolled into a mask. This can be seen in the figure 2 as the masks with the token IDs.

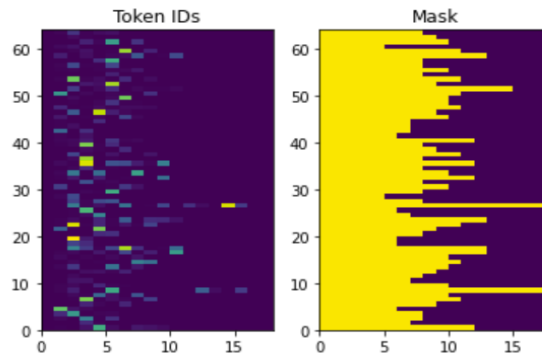


Figure 2:- mask and token id

VII. TRAINING FEATURES

Our model (NMT-1) was trained on the data collected through web scrapping which consist of the 3000 sentences of both from Hindi to English language. The limit on vocabularies to be the 1000 most common words for the two languages. Words do not present in these selected vocabularies are transformed into a worldwide token <UKG>. When teaching our Neural Machine Translator, subsequent (Bhadana, 2015), we clean out sentence pairs whose lengths exceed 40 words and mix up mini groups as we continue. Our assembling LSTM model have 3 layers, each with 250 cells, and 250-dimensional embeddings. The epochs to be set was 60. After the decoder function implementation collects the results into python lists, before using tf concat to join them into tensors. Our code is implemented in Jupyter notebook. we achieve a speed of 50 target words per second. It takes 4 hours to entirely train the model. Also, for comparison we trained the neural machine translation without attention. (NMT-2)

VIII. HINDI TO ENGLISH TRANSLATION ANALYSIS

We compare our Neural machine translation system with attention in the Hindi to English task with the neural machine translation without attention. It was interesting to examine the influence of attentional models in properly translating names such as “Rama” and “shiva”. Non-attentional models, while creating sensible names from a language model point of view, lack the direct contacts from the source side to make accurate translations.

IX. RESULTS

The neural machine translation system results acquired from NMT-1, NMT-2 and translations of Google and Bing systems, evaluated using BLEU score and can be shown in the table 1. Also, the attention plot for the attention-based model that is more focused on the diagonal can be shown in figure 3.

Translators	BLEU-1	BLEU-2	BLEU-3	AVERAGE
NMT-1	53.12	32.15	20.12	35.13
NMT-2	30.12	20.12	12.12	20.78
GOOGLE TRANSLATE	69.46	49.13	40.21	52.91
BING	59.56	40.23	26.45	41.49

Table1: - different BLEU score

So, looking on the score of bleu we can conclude that NMT-1 model which is LSTM built by directional encoder-decoder RNN with attention mechanism gives a better BLEU than the model without attention. The attention mechanism acknowledges components of the key sequence that are related to each word in the output and combine relevant information to select the correct output.

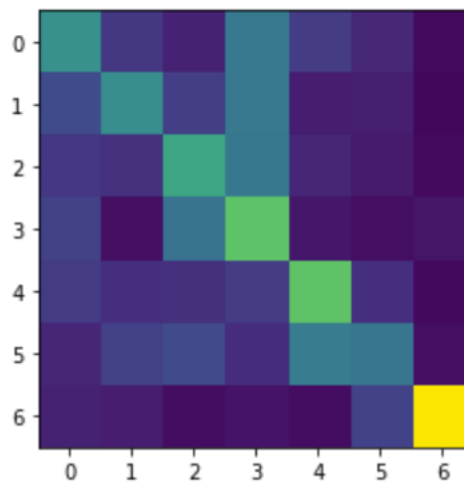


Figure3:- attention plot

X. CONCLUSION AND FUTURE WORKS

Even though, Neural machine translation system offers better accuracy than traditional methods like rule-based Machine translation as well as SMT, but it still delays behind equated to manual human translation. In this paper, NMT systems namely, sequence-to-sequence RNN with attention-based mechanism Hindi to English translation and matched with the existing MT output in terms of BLEU score. It shows up improved functioning than the present systems. Though, close analysis of calculated translations notes that our NMT systems need to be enhanced in case of identification of blank lines in output and different translation of the trace sentence. Besides, with the understanding of the impact of the bi-gram model in Hindi language translation and correlation among Indian similar languages in [7], opens a new direction of research for direct translation between pairs of similar languages.

XI. REFERENCES

- [1] Shachi Dave, Jignashu Parikh, and Pushpak Bhattacharyya, "Interlingual based englishhindi machine translation and language divergence," Machine Translation, vol. 16(4), pp.251–304 (2001).
- [2] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "Learning phrase representations using rnn encoderdecoder for statistical machine translation," In

Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, pp. 1724–1734 (2014).

[3] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le, “Sequence to sequence learning with neural networks,” In proceedings of the 27th international conference on neural information processing systems - Volume 2. MIT Press, Cambridge, MA, USA, NIPS14, pp. 3104–3112 (2014).

[4] Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya, “The IIT bombay english-hindi parallel corpus,” In proceedings of the eleventh international conference on language resources and evaluation (LREC 2018), May 7-12, 2018, Miyazaki, Japan, isbn: 979-10-95546-00-9, (2018).

[5] Amarnath Pathak and Partha Pakray, “Neural machine translation for Indian languages,” Journal of Intelligent Systems (JISYS), DOI: <https://doi.org/10.1515/jisys-2018-0065> (2018)

[6] Sainik Kumar Mahata, Dipankar Das, and Sivaji Bandyopadhyay, “Mti2017: Machine translation using recurrent neural network on statistical machine translation,” Journal of Intelligent Systems pp. 1–7 (2018).

[7] Sayan Ghosh, K V Vijay Girish and Tv Sreenivas, “Relationship between indian languages using long distance bigram language models,” In proceedings of ICON-2011: 9th international conference on natural language processing, Macmillan, pp. 104–113 (2011).

[8] Sahinur Rahman Iakar, Abinash Dutta, Partha Pakrey, Sivaji Bandyopadhyay, “Neural Machine translation English to hindi”