International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-4, November 2019

Sentence Wise Telugu to English Translation of Vemana Sathakam using LSTM

P.Sujatha, D. Lalitha Bhaskari

Abstract*: Language translation is a power of humans where machines are lagging and need to acquire. Previous statistical machine translation is used for translation but is applicable for large and similar grammar structure dataset. In this paper neural machine translation with long short term memory (LSTM) is used for addressing the issue. This paper uses a bidirectional LSTM to translate Telugu literary poems of Yogi Vemana to English which exhibited satisfactory translation. The results are compared with existing and proposed methods. NMT with LSTM yields better in language translation.

***Keywords* : Machine translation, Neural machine translation, Long Short Term Memory.**

I. INTRODUCTION

From past 50 years machine translation was one of the initial task in research field by computer scientists. It can also referred as automatic translation. Machine translation system mainly of three types namely rule based system, statistical translation system and neural machine translation. Rule based system uses grammar rules and combination of languages with dictionaries for common words. It uses manually created dictionaries with consistent translations to get accurate terminology. It is robust and gives high performance but needs high development and customization costs. Statistical systems doesn't follow any grammar rules. It can learn by itself by analysing amounts of data for each language pair. Not perfectly accurate when compared to other translation systems. It requires rapid and cost effective development costs provided the required corpus exists. Neural Machine Translation (NMT) is a new method which uses large neural network to learn and translate the given data. One of the popular approach among machine translators is NMT. Developers and researchers follow NMT for accurate translation performance in many language pairs when compared to other machine translator systems.

In recent years the biggest search engine Google also has focused on Neural Machine Translation for translation. Google has developed a new approach called GNMT (Google Neural Machine Translation). It is an eight layer encoder-decoder architecture. For training neural network it requires huge GPU computations. In this paper we have used LSTM architecture for translation and results were promising for the given language.

Manuscript published on November 30, 2019.
* Correspondence Author
P. Sujatha*, research scholar, Department of Computer Science and Systems Engineering, Andhra university college of engineering (A), Andhra university.
D. Lalitha Bhaskari, Professor, Department of Computer Science and Systems Engineering, Andhra university college of engineering (A), Andhra university.

II. MACHINE TRANSLATION

A. Neural Machine Translation

Neural machine translation is a new methodology for machine translation which uses artificial neural networks to increase the frequency and accuracy. It is a simple encoder-decoder network model for machine translation. The ultimate goal of neural machine translation system is takes a sentence from one language as input and translate that sentence into other language as output. For any machine translation system the first task is to convert textual data into a numeric form. If we want to convert any textual data into numeric form, we have to transform each word into one hot encoding vector which is given as input to the model. Every word has given an index starting with 0. So that each word can have a corresponding hot encoding vector thus we can represent our dataset with numerical representation. This process can be applied for both input and output languages.

Neural Machine Translation uses a network model called RNN (Recurrent Neural Network). It is the basic architecture for NMT. It is a cyclic structure which enables the learning of the repeated sequences are easy compared to other networks. It can be unrolled to keep sentences in a sequence manner in both sources and target languages. The architecture of RNN is depicted below in Fig.1.

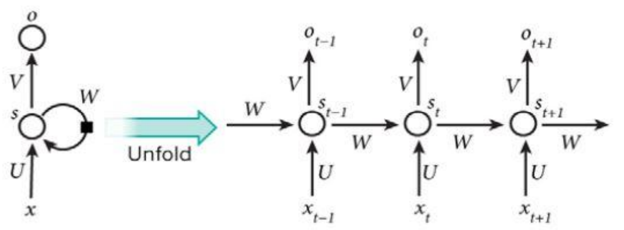


Figure 1. Architecture of RNN

This architecture describes how multiple layers can be unrolled from single layer. A single cell can be used to store an information of the previous time period. Mapping between sequences to sequences can be easy in RNNs if the alignment is known between inputs and outputs ahead of time. Let us assume source sentence as X and target sentence as Y. Source sentence x_1, x_2, x_3 …x_n can be converted by RNNs encoder into various vectors of fixed dimensions.

The decoder uses conditional probability to provide one word at a time as its output.

$$P(Y|X) = P(Y|X_1, X_2, X_3 \ldots X_m) \quad (1)$$

Where $X_1, X_2, X_3 \ldots X_M$ are the fixed dimensional vectors encoded by RNNs encoder. The above equation can be used convert to below equation by using chain rule.

The symbols predicted till now can be used to predict the next word and source sentence vectors. The converted equation can be expressed as

$$P(Y|X) = P(y_i|y_0, y_1, y_2 \ldots, y_{i1}; X_1, X_2, X_3 \ldots X_m) \quad (2)$$

A softmax function is used to represent the distribution of each term in the distribution of the vocabulary. RNNs work well with short sentences in theory but training with long sentences is a tough job for RNN because it is suffering from "Long Term Dependency Problem". The reason behind is that in RNN to find the decision at time t is defined as

$$\frac{\partial ht}{\partial h0} = \frac{\partial ht}{\partial ht-1} * \frac{\partial ht-1}{\partial ht-2} * \ldots \frac{\partial h2}{\partial h1} * \frac{\partial h1}{\partial h0} \quad (3)$$

The output of the long sentences is very low and the results are also not perfectly accurate because of multiplicative effect. Let us consider an example for clear understanding. By using a language model if we want to predict a next word in a sentence then in sentence like "Sun rises in the ________. As we know the answer is **east.** Because of the relevant information and its small place it is somehow easy to predict. For example "I am a student, I study well and I sits in the class in ________ bench. It is not easy to predict because of the recent information we can only predict that missing word may be a position in sitting in class. There are multiple choices like first, middle, last bench position. This problem can be clearly says that the gap between relevant information and its place is not small and it is somewhat difficult for the model to predict when compared to first example. To deal with such complex sentences and cases RNN may not be suitable for analysis. To overcome this problem LSTM (Long and Short Term Memory) comes into picture. LSTM works better for encoding and decoding over RNNs.

B. Long and Short Term Memory (LSTM) model

In machine translation networks some times where there are dependencies of long range temporal problems occur we go for a replacement of RNN with LSTM. The structure of LSTM is somewhat similar to structure of RNN. The number of layers used is the main difference, RNN uses a single layer whereas LSTM uses four layers. The LSTM also has the structure represented by a series of repeating modules. For the purpose of learning the layers interact with each other as well with the layers of other modules. The structure of LSTM is given in figure 2.

The cell state acts as a conveyor belt for some linear minor actions. The addition or removal of data to or from the cell state is regulated by structure known as gates is the ability of LSTM. There are four main gates in LSTM structure. The first gate is "forget gate layer" which is a sigmoid layer which decides what information to be sent for the cell state. The output value ranges from 0 to 1, in which 0 represents send no data to cell state, 1 represents send all data to cell state.

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \quad (1)$$

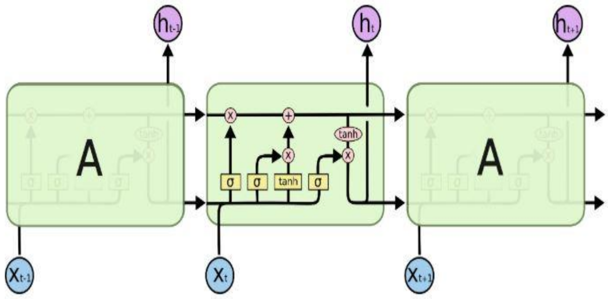


Figure 2. Structure of LSTM

The next gate has two layers, the first is "input gate layer" which is a sigmoid layer and the other is a tanh layer. The working of sigmoid layer is to decide which data to be updated. A new candidate value vector $\hat{C}_t$ is created by the tanh layer. Both the values from sigmoid layer and tanh layer are combined in the next step.

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \quad (2)$$

$$\hat{C}_t = tanh(W_c.[h_{t-1}, x_t] + b_c) \quad (3)$$

Now it's time to combine the outputs of the two layers i.e., forget gate layer and input gate layer to get a new cell state C_t.

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t \quad (4)$$

The final output is a filtered output based on cell state. Two steps are there to finalise the output. The first is a sigmoid layer used for deciding the output from cell state. The second is a tanh layer used to get results between -1 to 1 and are multiplied with outputs of sigmoid layer to decide the final output.

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = f_t * C_{t-1} + i_t * \hat{C}_t \quad (6)$$

Figure 3 represents the translation of a sentence ABC to produce output as WXYZ, stops after encountering <end>.

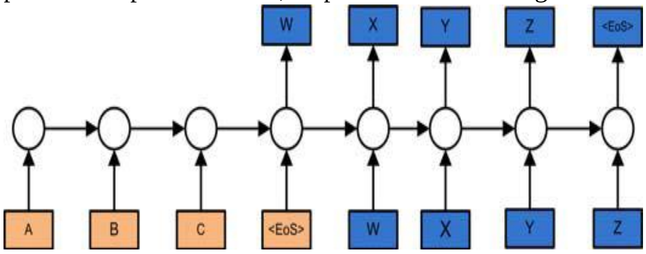


Figure 3. Sentence modeling in LSTM network

III. THE ESTABLISHMENT OF NEURAL MACHINE TRANSLATION SYSTEM

A. Encoder and Decoder

A bidirectional encoder is used in LSTM based NMT. The output of the encoder is context vector *ct*. The concept of encoder is that the output at an instant time depends on the both past and future data. The LSTM is tweaked so that the hidden layers from opposite directions can be connected to the same output. This version of tweaked LSTM is known as Bidirectional LSTM (Bi-LSTM). To increase the available input information to the network, Bi-LSTM is introduced. Bi-LSTM does not need time delays like LSTM but uses future input from present state. The below figure gives an overview about NMT which uses bidirectional encoder.

The figure for single layer encoder is given below. The GNMT was first introduced in November 2016 which was enabled for eight languages. To improve the efficiency multiple layers of LSTM can be used in both encoder and decoder designs.

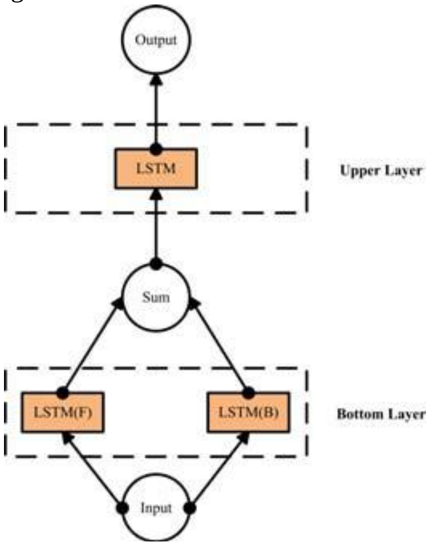


Figure 4. Bidirectional Encoder design using Bi-LSTM

To decode the context vector *ct* which is obtained after encoding the input is decoded back to the targeted sentences or words a decoder is designed. Multidirectional decoder system is used. A two layered decoder is shown in figure 5.

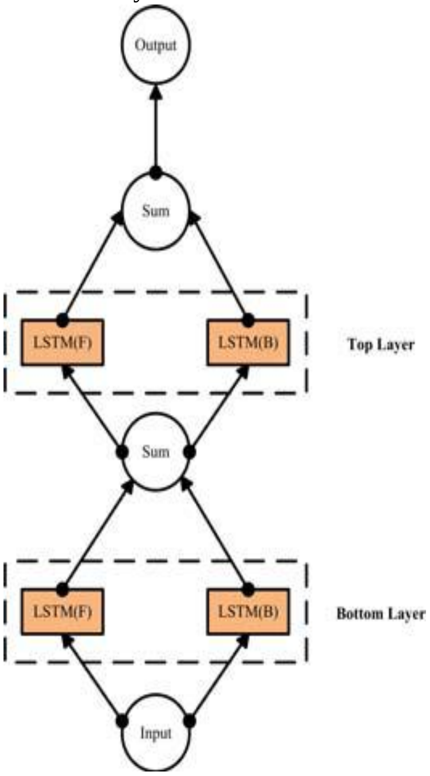


Figure 5. 2 layer decoder architecture

B. Attention in the model

Attention layer acts as an interface between encoder and decoder layers of an NMT system. The attention models are of two types. They are global and local. The main idea of the attention model is using of all the hidden encoder states while computing the context vector *ct*. An alignment vector which is variable-length whose length equals to the size of time steps count of the source side is calculated by comparison of current target hidden state (*ht*) and sourse hidden state (*hs*). In the local attention model the modelling concept of languages is different. In case of local attention model, it first predicts a single aligned position *pt* of the current target word. By using this position a window of words is determined which the model attends to. With the help of this window a context vector *ct* is computed using the window that is centered around source position *pt*.

Local attention model is used and the diagram showing functionality of local attention model is shown below. All words of the source side are attended for every target word is the downside of the global attention model which is expensive. A position is chosen to over this in local attention model.

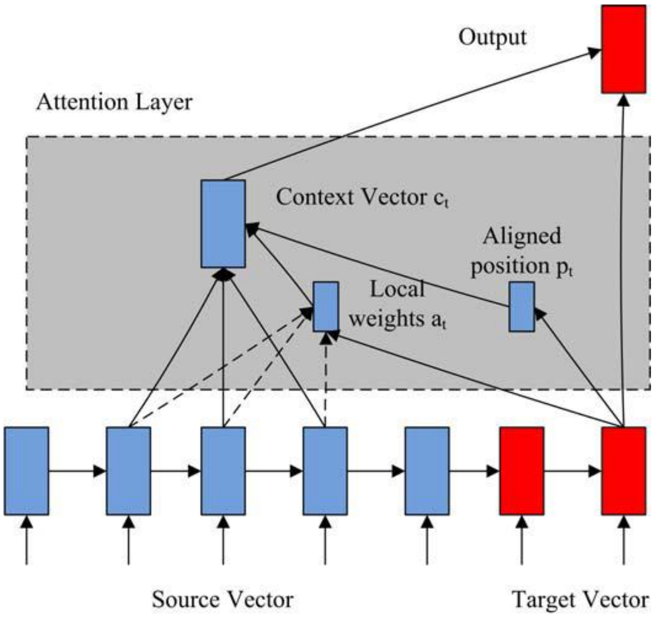
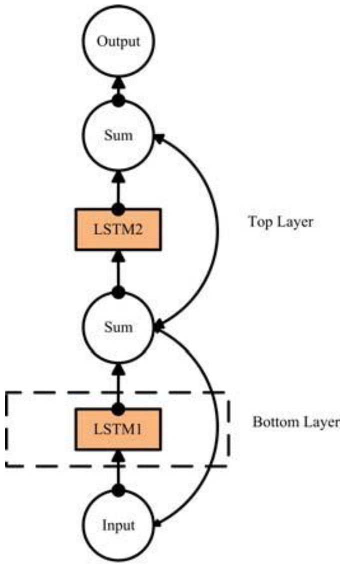


Figure 6. Local attention model

- Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.
- Do not mix complete spellings and abbreviations of units: “Wb/m2” or “webers per square meter”, not “webers/m2”. Spell out units when they appear in text: “. . . a few henries”, not “. . . a few H”.

C. Residual connections and Bridges

The depth of the network is mainly considered for the success of a neural network. However, it is difficult to train the network when the depth of the network increases because vanishing and exploding gradients. A model is generated for differences between outputs of intermediate layers and the target output that is used for addressing the gradient problem. These are known as residual connections. The input at a layer is added element wise with the output of previous layer before feeding it as input to the next layer. In below figure the output obtained from LSTM1 is added with the input and given as input to LSTM2. The gradient flow is greatly improved in the back propagation using residual connections, so that we can train deeper networks.



Encoder

Figure 7. Residual connections inside the Encoder

An additional attention layer is needed between encoder layer and decoder layer. The below figure shows the layout consisting of encoder, decoder, residual connections and bridge.

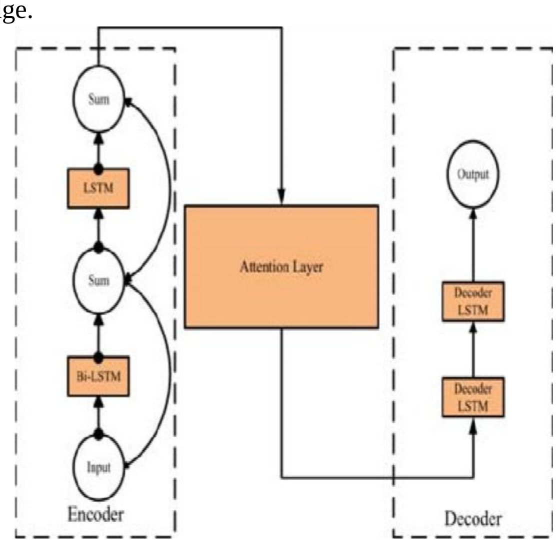


Figure 8. Complete system block diagram for NMT

IV. DATASET

The data set is collected from web as well as contains the data that I have added with the help of my project guide so as to get better results. The data set contains data regarding a few poems of Telugu literature and its translation in English. The data set mainly consists of Vemana poems.

V. RESULTS AND DISCUSSION

To maintain the fixed length input and output are converted to integer sequences in seq to seq model. But before we do that, let's visualise the length of the sentences. We will capture the lengths of all the sentences in two separate lists for English and Telugu, respectively.

Quite intuitive – the maximum length of the Telugu sentences is 10 and that of the English phrases is 15.

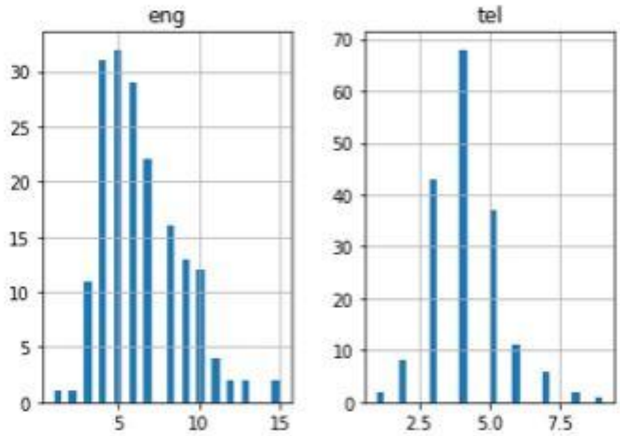


Figure 9. Visualization graph for sentence lengths

Telugu sentence is taken as input for translation to English. The translated English sentence is displayed along with given Telugu sentence.

```
Enter a telugu sentence:
గంగి గోవు పాలు గరిటడైనను చాలు
The corresponding English sentence is :
 a ladleful of a sacred cows milk is enough

Enter a telugu sentence:
ఉప్పు కప్పురంబు నొక్క పోలిక నుండు
The corresponding English sentence is :
 salt and camphor look similar

Enter a telugu sentence:
పురుషులందు పుణ్య పురుషులు వేరయ
The corresponding English sentence is :
 among men virtuous people stand apart
```

Figure 10. Output

Let's compare the training loss and the validation loss.

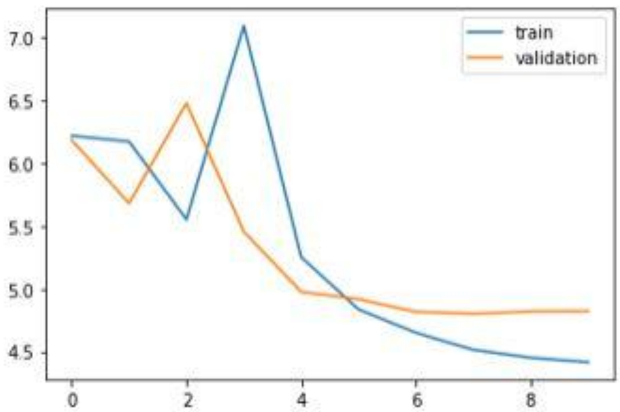


Fig 7.5: Train & Validation graph

VI. CONCLUSION

LSTM based NMT is used for solving the issue of accuracy as well as the need for large dataset in statistical machine translation. We have used a dataset with less data and the number of layers used is also less. The results have proven that LSTM with NMT gives better results with less amount of data both for training and validation, the results are satisfactory.

Future work could be extended by considering datasets like different sathakas, idioms, proverbs and holy books to translate Telugu to English and vice versa.

REFERENCES

1. Dzmitry Bahdanau and KyungHyun Cho Yoshua Bengio, "Neural Machine Translation By Jointly Learning to Align and Translate", Published as a conference paper at ICLR 2015
2. Sandeep Saini and Vineet Sahula, "Neural Machine Translation for English to Hindi", 2018 Fourth International Conference on Information Retrieval and Knowledge Management
3. Keerthi Lingam, E. Rama Lakshmi and L Ravi Theja, "Rule-based Machine Translation From English to Telugu With Emphasis on Prepositions", 2014 First International Conference on Networks & Soft Computing
4. B.N.V Narasimha Raju and M S V S Bhadri Raju," Statistical Machine Translation System for Indian Languages", 2016 IEEE 6th International Conference on Advanced Computing
5. Jayashree Nair, Amrutha Krishnan K and Deetha R, "An Efficient English to Hindi Machine Translation System Using Hybrid Mechanism", 2016 Intl. Conference on Advances in Computing, Communications and Informatics (ICACCI), Sept. 21-24, 2016, Jaipur, India
6. Rosna P Haroon and Shaharban T A, "Malayalam Machine Translation using Hybrid Approach", International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) 2016
7. Omkar Dhariya, Shrikant Malviya and Uma Shanker Tiwary, "A Hybrid Approach For Hindi-English Machine Translation", International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) 2017
8. Teddy Mantoro, Jelita Asian and Media A. Ayu," Improving the Performance of Translation Process in a Statistical Machine Translator Using Sequence IRSTLM Translation Parameters and Pruning", 2016 International Conference on Informatics and Computing (ICIC)
9. Hu Bing, "Statistical Machine Translation Algorithm Based on Improved Neural Network", International Conference on Robots & Intelligent System 2017
10. Kehai Chen, Tiejun Zhao, Muyun Yang, Lemao Liu, Akihiro Tamura, Rui Wang, Masao Utiyama and Eiichiro Sumita, "A Neural Approach to Source Dependence Based Context Model for Statistical Machine Translation", ACM Transactions on Audio, Speech and Language Processing, Vol 26, No 2, February 2018.
11. Sandeep Kharb, Hemant Kumar, Manoj Kumar and Dr. Arun Kumar Chaturvedi, *"Efficiency of a Machine Translation System"*, International Conference on Electronics, Communication and Aerospace Technology ICECA 2017
12. Pooja Malik and Anurag Singh Baghel, "An Algorithm for the better Assessment of Machine Translation", International Conference on Computing, Communication and Automation (ICCCA2017)
13. Mani Bansal and Goonjan Jain, "Improvement of English-Hindi Machine Translation using ConceptNet",Recent Developments in Control, Automation and Power Engineering(RDCAPE) 2017
14. Aditya Kaustav Das, Manabhanjan Pradhan, Amiya Kumar Dash, Chittaranjan Pradhan and Himansu Das, "A Constructive Machine Translation System for English to Odia Translation", International Conference on Communication and Signal Processing, April 3-5, 2018, India

AUTHORS PROFILE

Mrs.P Sujatha, pursued her MCA from Andhra University in the city of Visakhapatnam, India in 2004 and Master of Technology from GITAM College of Engineering, Vizag Campus in the year 2009. She is currently working as Research Scholar in the Department of Computer Science and Systems Engineering, Andhra University, India since April 2014. She has 5 years of teaching experience and her research interests are in the fields of Image Processing, Computer vision, Artificial Intelligence and Machine learning.

Prof. D. Lalitha Bhaskari, is working as professor in the department of Computer Science & Systems Engineering(CS&SE), AUCE(A), Andhra University, Visakhapatnam. She has 20 years of teaching experience and her research areas include cryptography & Network Security, Computer Vision, Cyber Forensics & Deep learning. She has nearly 50 papers in reputed international journals and 32 papers in proceedings of international conferences. She has done international consultancy works related to deep learning with respect tp image segmentation .She is a life member of a few reputed professional bodies. She is a receipient of "Young Engineers" award by IEI, India during the year 2009