International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Neural Technique for Language Translation

Arun P N[1], Harikesh B G[2], Rakesh S R[3], Sahithya A N[4], Jayanna H S[5]

Submitted: 10/09/2022 **Accepted:** 20/12/2022

Abstract : Objectives: To develop a Neural Machine Translator which can be integrated to the chatting applications which will be helpful for the users who are convenient with their regional languages. **Methods**: Neural machine translation is an approach in machine translation which uses an artificial neural network to predict the likelihood of a sequence of words. It is typically modeling entire sentences in a single integrated model. NMT provides more accurate translation by taking into account the context in which a word is used, rather than just translating each individual word on its own. **Findings**: We used LSTM to build our model and we were able to get the Hindi sentences for the corresponding English sentences which contains the words count less than are equal to 5 accurately. We were getting translation for sentences more than 5 words also but not all. Like if we test for 100 sentences having more than 5 words, we got almost 75 to 80 sentences accurately.

Keywords: *Neural Machine Translation, Keras, Recurrent Neural Network, LSTM, Encoder and Decoder.*

1. Introduction

Machine interpretation is the cycle of naturally changing over source text to objective content which are in various dialects. Programmed or machine interpretation is perhaps the most difficult AI undertakings given the ease of human language. At first standard based frameworks were utilized to achieve this errand. These standard based frameworks were supplanted by factual strategies in the last part of the 1990s. Lately profound neural organizations accomplish the necessary outcomes which are named as neural machine interpretation.

Machine Translation Approaches

1. Rule Based Systems: Rule-based knowledge otherwise called word-based interpretation is a methodology where the machine creates pre-characterized results dependent on specific principles coded by people. It applies those guidelines to store sort and control the information. It comprises of 4 segments.

   i. Set of rules which are known as rule-base.
   ii. Interference motor is known as a semantic reasoner. This deciphers the principles and makes a move likewise.
   iii. Temporary working memory. Utilizing this the interface motor executes the creation framework program.
   iv. A UI, permitting people to cooperate.

Advantages of Rule-based systems:

i. Accessibility: Availability of the framework for the client isn't an issue
ii. Cost effective: This framework is cost productive and exact as far as its outcome
iii. Speed: You can improve the framework as you probably are aware every one of the pieces of the framework. So, to give yield shortly is definitely not a major issue
iv. Precision and less mistake rate: Although inclusion for various situations is less, whatever situations are covered by the RB framework will give high exactness. In view of these predefined rules, the mistake rate is likewise less
v. Diminishing danger: We are lessening the measure of hazard as far as framework exactness
vi. Consistent reaction: Output which has been created by the framework is reliant upon rules so the yield reactions are steady, which implies it can't be dubious
vii. A similar psychological interaction as a human: This framework gives you a similar outcome as a human, as it has been hand tailored by people

Dis-advantages of Rule-based systems:

i. Parcel of manual work: The RB framework requests profound information on the space just as a great deal of manual work
ii. Tedious: Generating rules for an unpredictable framework is very difficult and tedious
iii. Less learning limit: Here, the framework will create the outcome according to the standards so the learning limit of the framework without help from anyone else is substantially less
iv. Complex spaces: If an application that you need to construct is excessively perplexing, building the RB framework can take part of time and examination.

[1] *Siddaganga Institute of Technology (SIT), Tumkuru- 572103, Karnataka India. ORCID ID: 0000-0002-2423-9302*
[2] *Siddaganga Institute of Technology (SIT), Tumkuru- 572103, Karnataka India. ORCID ID: 0000-0002-9246-2012*
[3] *Siddaganga Institute of Technology (SIT), Tumkuru- 572103, Karnataka India. ORCID ID: 0000-0002-6154-6416*
[4] *Siddaganga Institute of Technology (SIT), Tumkuru- 572103, Karnataka India. ORCID ID: 0000-0002-3037-632X*
[5] *Siddaganga Institute of Technology (SIT), Tumkuru- 572103, Karnataka India. ORCID ID: 0000-0002-4342-9339.*
* *Corresponding Author Email: arunpn866@email.com*

Complex example recognizable proof is a difficult assignment in the RB approach

2. Statistical Machine Translation: Factual machine interpretation otherwise called express based interpretation is a methodology where it figures out how to decipher dependent on the examination of existing human interpretations. They gather interpretations utilizing cross-over phrases. In state-based interpretation, the point is to diminish the limitations of word-based interpretation by deciphering entire arrangements of words, where the lengths might contrast. The groupings of words are called phrases, yet normally are not etymological expressions, but rather expresses discovered utilizing measurable strategies from a bilingual book corpus. Factual machine interpretation uses measurable interpretation models whose boundaries originate from the examination of monolingual and bilingual corpora. Building factual interpretation models is a fast cycle, yet the innovation depends vigorously on existing multilingual corpora. Figure 1, depicts the block diagram of Statistical Machine Translation.

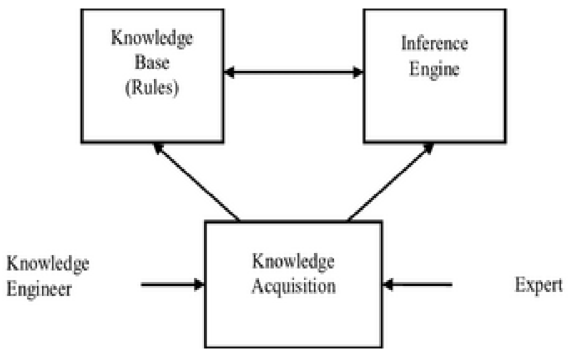

Figure 1: Statistical Machine Translation

Advantages of statistical machine translation:

i. Good fluency.
ii. Good catching exceptions to rules.
iii. Rapid and cost-effective development costs provided the required corpus exists.

Dis-advantages of statistical machine translation:

i. It doesn't perform well when the translating input is not similar to the training data.
ii. These systems need bilingual content. It will be tricky when it comes to finding content written in rarer languages.
iii. It is expensive. Preprocessing and corpus creation is not only expensive and time-consuming, but it also requires collaboration with computer scientists, translators and linguists.
iv. Once they're implemented, it's harder to fix mistakes in the system.

Rule Based MT vs. Statistical MT

Rule-based machine interpretation gives great out-of-area quality and is naturally unsurprising. Word reference-based customization ensures worked on quality and consistence with corporate phrasing. Be that as it may, interpretation results might come up short on the familiarity per users anticipate. As far as venture, the customization cycle expected to arrive at the quality edge can be long and expensive. The exhibition is high even on standard equipment.

Measurable based machine interpretation gives great quality when enormous and qualified corpora are free. The interpretation is familiar, which means it understands well and consequently meets client assumptions. In any case, the interpretation is neither unsurprising nor steady. Preparing from great corpora is computerized and less expensive. Be that as it may, preparing on broad language corpora, which means text other than the predetermined space, is poor. Besides, measurable MT requires huge equipment to construct and oversee huge interpretation models.

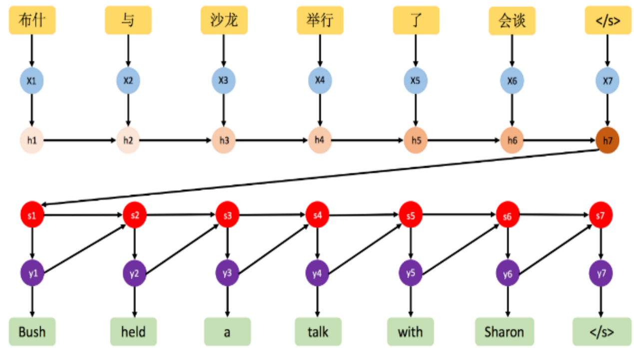

Figure 2: Neural Machine Translation

3. Neural Machine Translation: Neural machine[4][5] interpretation depends on the model of neural organizations in the human mind. Data will go through various layers of neural organizations and produce the ideal yield. It utilizes profound learning methods to help itself to interpret text dependent on existing measurable models. NMT[7][8] can make more exact and quicker interpretations than SMT and has the capacity to create greater yield. Figure 2, depicts Neural Machine Translation[12][13].

Advantages of Neural Machine Translation:

i. Gives most extreme advantage and consolation to NMV clients, like actual isolation (assurance).
ii. No need of formal guidelines.
iii. Can be presented rapidly.
iv. Typically, somewhat modest.
v. No need of implementation.

Dis-advantages of Neural Machine Translation[18]:

i. Require guidelines, which might require public meetings, and in this way require some investment to execute.
ii. Requires authorization assets to guarantee consistence.
iii. Can be harder to configuration because of more elevated level of limitation on access by different vehicles.
iv. Such measures are generally just demonstrative, so may not generally be powerful.

Objectives

i. The main objectives for this paper work are as follows:
ii. Collect the data i.e., English sentences and find the corresponding Hindi sentences and perform pre-processing in order to train and test the model.
iii. Compare and analyze the various approaches so as to obtain an optimal and suitable approach.

iv. Build the encoder, decoder and neural network model to train and test from collected pre-processed data.
v. Once, the model gets trained correctly and produces appropriate output, test the model with different data sets.

2. Methodology

2.1. Proposed System

From the assets we alluded, we got an unmistakable picture that utilization of Long Short-Term Memory, likewise called as LSTM[16][19] will be a superior methodology for building a model for language interpretation.

We isolated our work into five phases. They are,

i. Data collection
ii. Data pre-processing
iii. Encoding and Decoding
iv. Training Model
v. Testing Model

2.2. System Design

The system design and the data flow diagrams are shown in this section:

As we examined in Proposed framework segment of Literature Survey section, we partitioned our working into five phases as displayed in figure 3.

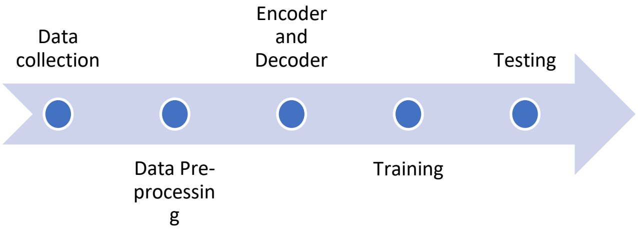

Figure 3: Stages to carry out the paper

The gathered information will be sifted and refined utilizing Python libraries like pandas. This refined information will be gone through different layers of a neural organization in encoder. Then, at that point yield of encoder will be passed into a decoder which will yield the end-product. After this, the prepared model will be utilized to check the yields utilizing the test information to confirm the rightness of model. High level system design is represented in figure 4.

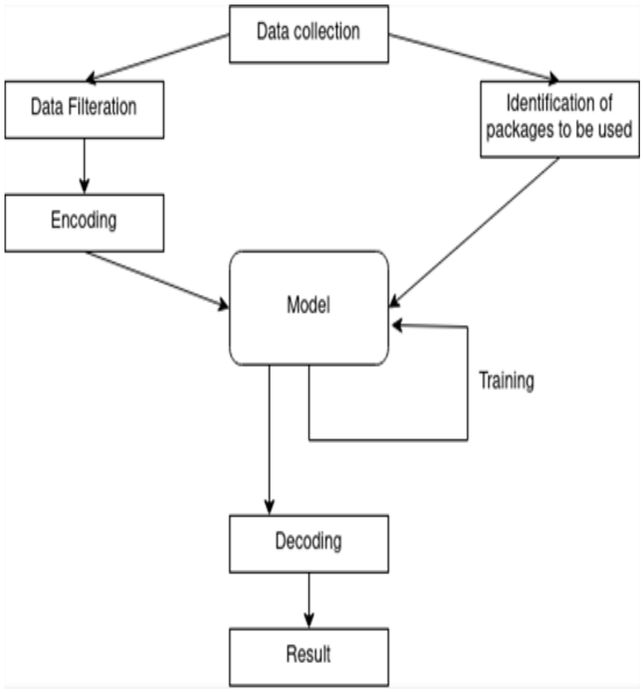

Figure 4: System Design

The uncleaned information will be pre-prepared utilizing a portion of the python libraries like pandas. Then, at that point, valuable information like word vocabulary, Number of encoder input tokens and Number of decoder output tokens will be registered and can be utilized while executing encoder, decoder, model. It is depicted in figure 5.

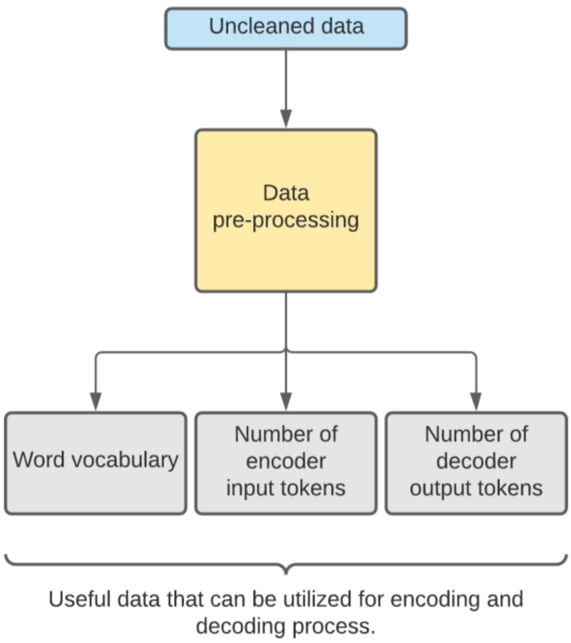

Figure 5: Data refining

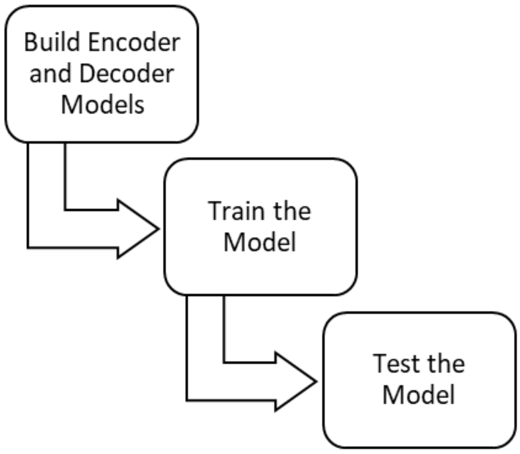

Figure 6: Path after data pre-processing.

The square graph in the figure, Figure 6, gives a way to continue further, when information pre-preparing is finished. After the pre-preparing of information is done, we constructed Encoder and Decoder. The pre-handled information is gone through Encoder and the encoded information is utilized for training.

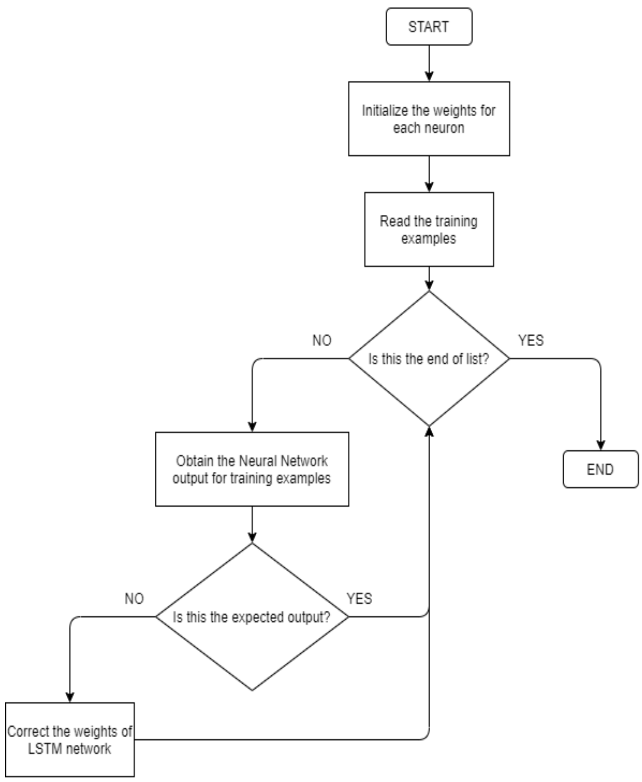

Figure 7: Flow diagram for training the model.

The figure, Figure 7, gives a reasonable image of Training the model. The preparation begins by introducing some arbitrary loads to every neuron. Then, at that point perusing some preparation models from the huge dataset which is pre-handled and encoded utilizing One-hot vectorization. Then, at that point the control will check whether it has arrived at the finish of rundown. Assuming indeed, Training closes and the loads that are applied on neurons will be put something aside for additional interaction. Assuming no, then, at that point the subsequent stage will acquire the Neural Network yield for preparing models. Presently the control checks for the accuracy of acquired yield with the normal one. Assuming indeed, the interaction rehashes from checking the finish of rundown. Assuming no, then, at that point the loads get refreshed and the interaction rehashes from checking the finish of the rundown.

2.2. System Implementation

Neural machine interpretation is the way toward changing over an arrangement of words from a source language to an objective language. For making an interpretation of starting with one language then onto the next, NMT[11][14] is considered as the most remarkable methodology.

A Convolutional Neural Network (ConvNet/CNN)[17] is a Deep Learning calculation which can take in an information picture, appoint significance (learnable loads and inclinations) to different perspectives/objects in the picture and have the option to separate one from the other. It is a particular neural organization that measures the information having shape like 2D framework. At the point when we are working with successive information that is needed to be endured throughout a few time steps, Recurrent Neural organization (RNN)[1] will be utilized. Figure 8 gives a pictorial example of CNN.

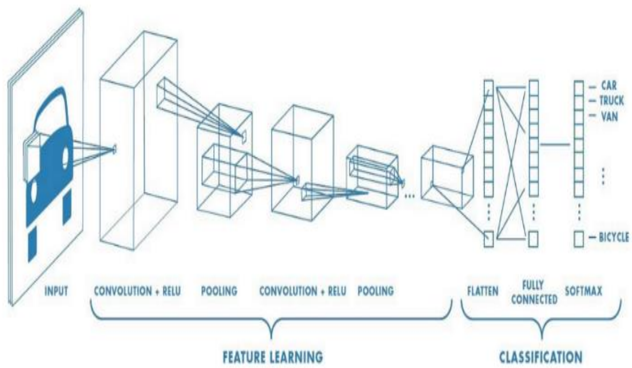

Figure 8: Example for Convolutional Neural Network.

Suppose we are composing a message "How about we meet for___ "and we need to foresee what might be the following word. The following word could be lunch, or supper or breakfast or espresso. It is simpler for us to make derivations dependent on the unique situation. Suppose in the event that we realized that we were meeting in the early evening and that data persevered in our memory then we can without much of a stretch make expectation that we are perhaps meeting for lunch.

At the point when we need to deal with successive information that should be endured throughout a few time steps then we utilize Recurrent Neural organization (RNN)[6]. Conventional neural organization and CNN's need a fixed info vector, apply actuation work on fixed arrangement of layers to create a fix measured yield.
RNN's are neural organizations with circles to continue data. RNN[10][15] are known as a repetitive as they play out similar assignment for each component in the succession and yield components are subject to past components or states. Figure 9 gives a pictorial example of RNN.

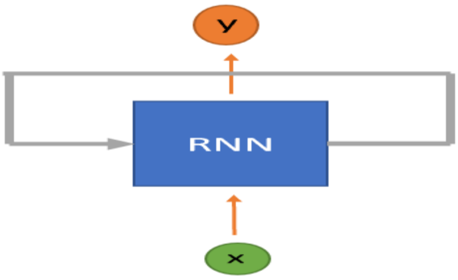

Figure 9: Example for Recurrent Neural Network.

Steps to train an RNN:

i. In the info layers, the underlying information is sent with all having a similar weight and actuation work.
ii. Using the current info and the past state yield, the present status is determined.
iii. Now the present status ht will become ht-1 for the subsequent time step.
iv. This continues rehashing for every one of the means, and to tackle a specific issue, it can go on as ordinarily to join the data from every one of the past advances.
v. The last advance is then determined by the present status of the last state and any remaining past advances.
vi. Now a mistake is produced by ascertaining the distinction between the real yield and the yield created by our RNN model.
vii. The last advance is the point at which the interaction of backpropagation happens wherein the mistake is backpropagated to refresh the loads.

Algorithm

1. The initial step prior to getting into the genuine center piece of the execution is information assortment and information pre-preparing.
2. The info and yield successions will be available in CSV design that can be separated for preprocessing utilizing pandas. The separated information will be isolated into two sections, one containing English successions and the other having Hindi arrangements. These successions will be first changed over into lower case and afterward pointless characters are taken out from the groupings. After the successions of the two dialects are separated, each arrangement is parted into singular words in order to fabricate a jargon of words. To store these words in both the dialects, set in python will be utilized. These arrangements of words in both the dialects go about as their particular vocabularies. We need to have the quantity of tokens in both the dialects which can be acquired by the length of vocabularies of the two dialects.

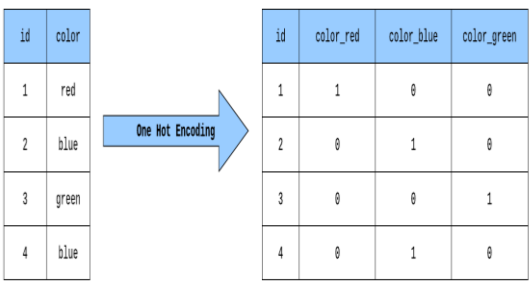

Figure 10: One hot encoding

3. Figure 10 shows how One hot encoding will be done. One hot encoding ought to be performed in the wake of setting up the word vocabularies. One hot encoding is the way toward distributing a remarkable number to every single novel symbolic present in the jargon of the language. As displayed in the figure over, each shading present in the rundown is addressed as a segment and the comparing section is addressed as paired 1 if the individual tone is available in that column. Similarly, the words or tokens present in the jargon is imagined as the segments and the crossing point the line and section of a symbolic will be addressed with paired 1 and rest of the segments present in that line will be addressed with parallel 0. Along these lines, an extraordinary portrayal for each symbolic present in the jargon should be possible utilizing one hot encoding.
4. The dataset will be isolated into two sections in which one of parts will be utilized for preparing the model and rest of the part will be utilized for testing the model. Clumps of certain size will be created utilizing the preparation part and the model will be prepared hence. There are two phases, one is the change of info arrangement into a vector known as encoding and the subsequent stage is the transformation of acquired resultant vector into the intelligible language known as disentangling.

Encoding

Encoding is the way toward changing over the tokens from intelligible language into vectors. We can utilize a portion of the Keras[20] classes to set up an encoder. From the start, the Input() class will return a Keras tensor (known as n-dimensional exhibit numerically). We can pass the necessary shape into the Input() class so it will return the tensor of that particular measurement. The got Keras tensor will be passed into Embedding() class which will take three sources of info,

i. Number of particular words in the preparation set.
ii. Size of the inserting vectors.
iii. Input length.

The Embedding() class yields the same loads for every one of the conceivable the information sources and these loads can be utilized for preparing the model. The loads acquired from the installing layer will be passed into a LSTM() class to set up an Encoder LSTM[9].

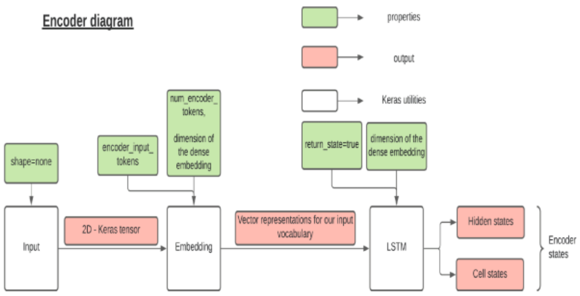

Figure 11: Implementation of Encoder

Figure 11 shows the block diagram of Encoder of our proposed system where mainly three blocks are used. Green block is properties applied on white box Keras utilities which produces pink box output.

LSTM

LSTM, in short for Long Short-Term Memory is a sort of repetitive neural organization that is fit for learning request conditions in grouping expectation type situations.

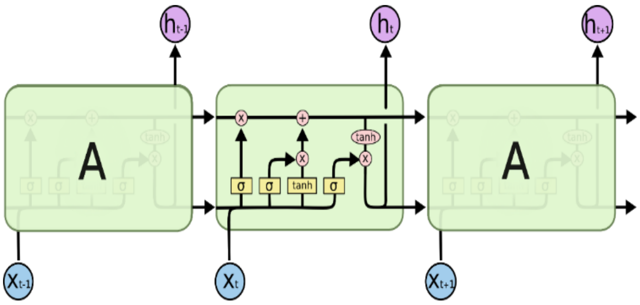

Figure 12: Long Short-Term Memory

The above figure, Figure 12, addresses the interior working of LSTM. Every cell addresses a layer of neural organization that gives covered up states, cell states and cell yields. These states can be utilized while arrangement expectations.

The figure, Figure 13, addresses a solitary cell of LSTM. There are two documentations C that addresses cell states and h that addresses covered up states. C(t-1) and h(t-1) address the covered up and cell conditions of past cell. C(t) and h(t) address the covered up and cell conditions of the current cell.

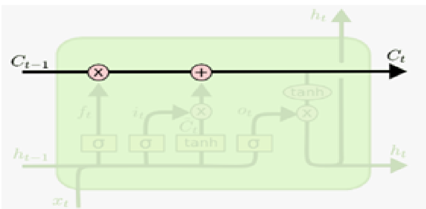
Figure 13: Representation of a Single cell in LSTM

Decoding

A comparable cycle will be conveyed for setting up a decoder that changes over the halfway vectors into intelligible yield. From the outset, a vacant Keras tensor will be introduced utilizing Input() class. This tensor for decoder will be passed alongside idle measurement as boundaries into an Embedding() class to get the loads required. The decoder

LSTM will be ready by passing idle measurement as a boundary and by making return_state and return_sequences as True, initial_state as encoder states so it will return the necessary yields. A thickly associated neural organization layer can be acquired from Dense() class in Keras by passing the quantity of decoder tokens as boundary. Thick layer is the standard profoundly associated neural organization layer. It is generally normal and often utilized layer. The decoder yields got from the decoder installing layer will be passed into the thick layers to acquire the last decoder yields.

Finally, the encoder inputs, decoder inputs, decoder yields will be utilized in the Model() class to prepare and create the model. A python work called generate_batch() has been carried out that will take training_data and batch_size as boundaries. This capacity will separate the preparation information into different groups and perform required tasks and afterward return the acquired yields at every cycle.

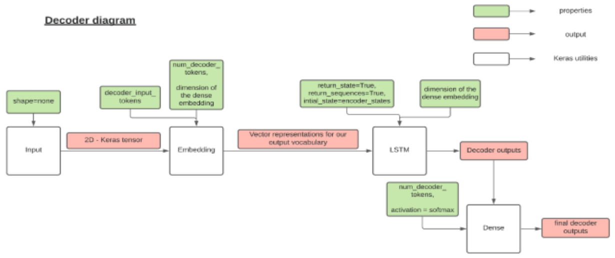

Figure 14: Implementation of Decoder

Figure 14 shows the block diagram of Decoder of our proposed system where mainly three blocks are used. Green block is properties applied on white box Keras utilities which produces pink box output. Though the block diagram seems to be similar to Encoder block diagram, the additional dense used in decoder plays a major role.

After the model is being prepared, another piece of the dataset which was put something aside for testing the model will be utilized to test the model's usefulness. A module called BLEU[2][3] score can be utilized to test the model's exactness by passing the necessary boundaries.

3. Results and Discussion

3.1. Training

While preparing the model, we took very nearly 25,000 sets of English-Hindi sentences. We prepared the model with those arrangement of sentences and the model was prepared for a standard worth of age, i.e., 100, and with a clump size of 128 for every age. The model starts preparing by executing some terminal lines of yields as displayed in figure beneath.

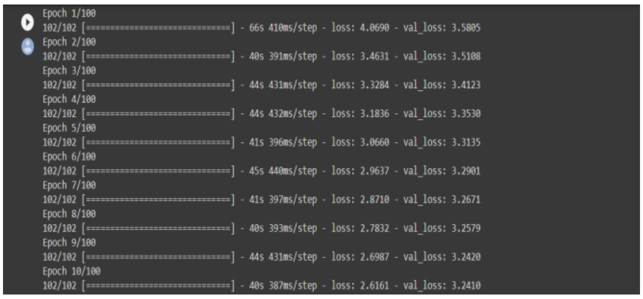

Figure 15: Some terminal output of training model

The figure, Figure 15, simply shows the yield for initial 10 Epochs and its comparing time term took to execute every age followed by time span for each progression. Like this, we can see the yield up to 100 ages. For every age, we can see the time length that it took for execution. Assuming the each of the 100 ages executed effectively with no interference, we will consider that the model is prepared and it is prepared for testing.

3.2. Testing

For testing, we separated the sentences as follows:

3-word sentences
4-word sentences
5-word sentences and so on

Essentially, we tried our model with the diverse length of sentences as referenced previously. Here, 3-word sentences mean, every one of the sentences with length three, i.e., it contains just three words. In like manner, we tried for 4-word sentences and 5-sentences additionally in this undertaking testing method. Assuming we need, we can test with 6-word, 7-word sentences and so on

In reality, for the estimation of proficiency of the prepared model, we utilized the idea of Bilingual Evaluation Understudy score, additionally called as BLEU score. For this, the primary concern is the loads we utilized while preparing. Here, we utilized four distinctive arrangements of loads.
The arrangement of loads utilized are,
(1, 0, 0, 0), (0.5, 0.5, 0, 0), (0.33, 0.33, 0.33, 0) and (0.25, 0.25, 0.25, 0.25)
For each set of loads utilized, we get distinctive BLEU score. Allow us to see those exhaustively.

The accompanying figures has three regions. Data English sentence (The English sentence we give as an information), Actual Hindi Translation (The Hindi sentence we are expecting to be for the particular English sentence) and Predicted Hindi Translation (The Hindi sentence that our model gives after understanding).

3-word sentences: 3-word sentences mean the sentences which contains just three words, i.e., the length of sentence ought to be three. The accompanying figure, Figure 16, shows the case of testing the interpretation in 3-word sentence.

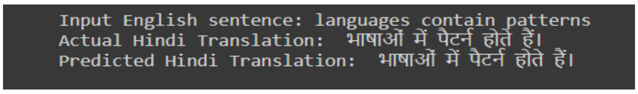

Figure 16: 3-word sentence testing

4-word sentences: 4-word sentences mean the sentences which contains just four words, i.e., the length of sentence ought to be four. The accompanying figure, Figure 17, shows the case of testing the interpretation in 4-word sentence.

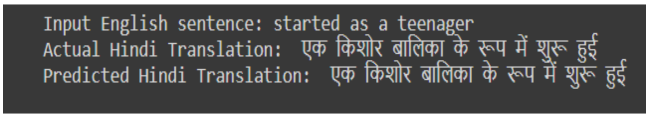

Figure 17: 4-word sentence testing

5-word sentences: 5-word sentences mean the sentences which contains just five words, i.e., the length of sentence ought to be five. The accompanying figure, Figure 18, shows the case of testing the interpretation in 5-word sentence.

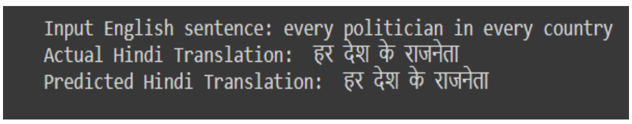

Figure 18: 5-word sentence testing

Furthermore, we likewise tried the sentences with arbitrary length, i.e., the sentences with in excess of 5 words. The accompanying figure, Figure 19, shows the case of testing the interpretation of a sentence with in excess of 5 words.

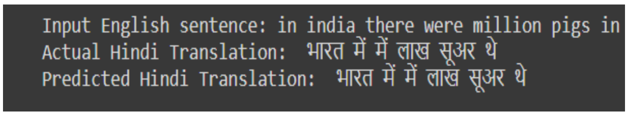

Figure 19: Testing sentence with more than 5 words

Sentence length	BLEU scores for particular weights			
	(1,0,0,0)	(0.5,0.5,0,0)	(0.33,0.33,0.33,0)	(0.25,0.25,0.25,0.25)
3	0.956522	0.978019	0.985438	0.988949
4	0.964286	0.981981	0.988070	0.990949
5	0.928571	0.963624	0.975841	0.981644
>5	0.945946	0.972598	0.981892	0.986204

Table 1: BLEU scores for tested sentences.

Like in the figure, Figure 19, we can test the model with numerous quantities of sentences with various number of words. According to our outcome investigation, our model predicts all sort of sentences shifted with number of words. However, a few sentences in uncommon case, our model predicts a few pieces of the gave input sentence. We can see the diverse BLEU score for various length of sentences that we tried against our prepared model. The Table 1 gives a reasonable picture that how productive our model predicts the yield.

4. Conclusion

In this paper, we showed the improvement of English-Hindi MT utilizing LSTM. We tried the created Engines with various classifications of sentences dependent on their length. For this, we did both human and programmed assessment. In the programmed assessment, we found that BLEU was creating better outcomes as displayed in Table 1, which was an improvement over the Baseline Model. In this section we reach determinations dependent on this paper and give a progression of errands that can be done in future to work on the framework.

In this task, we have applied neural machine interpretation method to decipher the local language like Hindi to English and the other way around. We began preparing the LSTM with the enormous dataset comprising of Hindi and English sentence sets. The accomplishment of our straightforward LSTM-put together methodology with respect to machine interpretation recommends that it ought to excel on numerous other grouping learning issues, if they have sufficient preparing information.

We have tried our machine interpretation framework both physically and naturally. In programmed assessment, we found that BLEU was creating better outcomes. We were additionally effectively ready to interpret long sentences from Hindi to English with high precision. Thus, further work will probably prompt much more noteworthy interpretation correctnesses. These outcomes recommend that our methodology will probably excel on other provoking succession to arrangement issue.

References

[1] Aqlan, F., Fan, X., Alqwbani, A., & Al-Mansoub, A. (2019). Arabic –Chinese neural machine translation: Romanized

Arabic as subword unit for Arabic - sourced translation. IEEE Access, 7, 133122-13313chrome5 [10.1109/ACCESS.2019.2941161].
[2] Revanuru, Karthik, Kaushik Turlapaty, and Shrisha Rao. "Neural machine translation of indian languages." Proceedings of the 10th Annual ACM India Compute Conference. 2017 [10.1145/3140107.3140111].
[3] Chauhan, T., and S. Sonawane. “The Contemplation of Explainable Artificial Intelligence Techniques: Model Interpretation Using Explainable AI”. International Journal on Recent and Innovation Trends in Computing and Communication, vol. 10, no. 4, Apr. 2022, pp. 65-71, doi:10.17762/ijritcc.v10i4.5538.
[4] Gehring, J., Auli, M., Grangier, D., Yarats, D., & Dauphin, Y. N. (2017). Convolutional sequence to sequence learning. arXiv preprint arXiv: 1705.03122 [1705.03122].
[5] Clinchant, S., Jung, K.W. and Nikoulina, V., 2019. On the use of BERT for neural machine translation. arXiv preprint arXiv:1909.12744 [1909.12744].
[6] Luong, Minh-Thang, Hieu Pham, and Christopher D. Manning. "Effective approaches to attention-based neural machine translation." arXiv preprint arXiv:1508.04025 (2015) [1508.04025].
[7] Snelleman, Emanuel. "Decoding neural machine translation using gradient descent." Master's thesis, 2016.
[8] Gain, Baban, Dibyanayan Bandyopadhyay, and Asif Ekbal. "IITP at WAT 2021: System description for English-Hindi Multimodal Translation Task." arXiv preprint arXiv:2107.01656 (2021) [2107.01656].
[9] Verma, C., A. Singh, S. Seal, V. Singh, and I. Mathur. "Hindi-English neural machine translation using attention model." International Journal of Scientific and Technology Research 8, no. 11 (2019): 2710-2714.
[10] Ghazaly, N. M. . (2022). Data Catalogue Approaches, Implementation and Adoption: A Study of Purpose of Data Catalogue. International Journal on Future Revolution in Computer Science &Amp; Communication Engineering, 8(1), 01–04. https://doi.org/10.17762/ijfrcsce.v8i1.2063
[11] Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun et al. "Google's neural machine translation system: Bridging the gap between human and machine translation." arXiv preprint arXiv:1609.08144 (2016) [1609.08144].
[12] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014) [1409.0473].
[13] Shah, Parth, and Vishvajit Bakrola. "Neural Machine Translation System of Indic Languages-An Attention based Approach." In 2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP), pp. 1-5. IEEE, 2019 [10.1109/ICACCP.2019.8882969].
[14] Zhang, Jingyi, Masao Utiyama, Eiichro Sumita, Graham Neubig, and Satoshi Nakamura. "Guiding neural machine translation with retrieved translation pieces." arXiv preprint arXiv:1804.02559 (2018) [1804.02559].
[15] Chawla, A. (2022). Phishing website analysis and detection using Machine Learning. International Journal of Intelligent Systems and Applications in Engineering, 10(1), 10–16. https://doi.org/10.18201/ijisae.2022.262
[16] Luong, Minh-Thang, and Christopher D. Manning. "Stanford neural machine translation systems for spoken language domains." In Proceedings of the international workshop on spoken language translation, no. IWSLT. 2015.
[17] Premjith, B., M. Anand Kumar, and K. P. Soman. "Neural Machine Translation System for English to Indian Language Translation Using MTIL Parallel Corpus." Journal of Intelligent Systems 28, no. 3 (2019): 387-398 [10.1515/jisys-2019-2510].
[18] Datta, Debajit, Preetha Evangeline David, Dhruv Mittal, and Anukriti Jain. "Neural machine translation using recurrent neural network." International Journal of Engineering and Advanced Technology 9, no. 4 (2020): 1395-1400.
[19] Using a Keras Long Short-Term Memory (LSTM) Model to Predict Stock Prices by Derrick Mwiti, Data Analyst. (https://www.kdnuggets.com/2018/11/keras-long-short-term-memory-lstm-model-predict-stock-prices.html)
[20] CNN computation over RNN by FaceBook (https://engineering.fb.com/2017/05/09/ml-applications/a-novel-approach-to-neural-machine-translation/)
[21] Beam search and Attention Mechanism (https://hackernoon.com/beam-search-attention-for-text-summarization-made-easy-tutorial-5-3b7186df7086)
[22] LSTM (http://colah.github.io/posts/2015-08-Understanding-LSTMs/)
[23] Keras (https://blog.keras.io/a-ten-minute-introduction-to-sequence-to-sequence-learning-in-keras.html)
[24] M. J. Traum, J. Fiorentine. (2021). Rapid Evaluation On-Line Assessment of Student Learning Gains for Just-In-Time Course Modification. Journal of Online Engineering Education, 12(1), 06–13. Retrieved from http://onlineengineeringeducation.com/index.php/joee/article/view/45